? logon

```
*** It is now 2010/02/24 08:05:34 ***
 (Dialog time 2010/02/24 08:05:34)
```

? b 411

```
       24feb10 08:06:03 User276702 Session D274.1
            $0.00    0.251 DialUnits File415
     $0.00  Estimated cost File415
     $0.14  INTERNET
     $0.14  Estimated cost this search
     $0.14  Estimated total session cost   0.251 DialUnits
File 411:DIALINDEX(R)

DIALINDEX(R)
   (c) 2010 Dialog

*** DIALINDEX search results display in an abbreviated ***
*** format unless you enter the SET DETAIL ON command. ***
```

? sf all

```
   You have 508 files in your file list.
   (To see banners, use SHOW FILES command)
```

? s (semiadditive? or (semi () additive)) (n2) (measure? or metric? or value? or parameter?) and (cube? or account? or microsoft) not py>2004

Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing

Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing

Processing
Processing
Processing
Processing
Processing

```
Your SELECT statement is:
   S (SEMIADDITIVE? OR (SEMI () ADDITIVE)) (N2) (MEASURE? OR METRIC? OR
VALUE? OR PARAMETER?) AND (CUBE? OR ACCOUNT? OR MICROSOFT) NOT PY>2004

           Items   File
           -----   ----
               1    15: ABI/Inform(R)_1971-2010/Feb 23
       Examined  50 files
       Examined 100 files
       Examined 150 files
       Examined 200 files
               1   324: GERMAN PATENTS FULLTEXT_1967-201003
               1   349: PCT FULLTEXT_1979-2010/UB=20100205|UT=20100204
       Examined 250 files
       Examined 300 files
       Examined 350 files
       Examined 400 files
Processing
Processing
               2   654: US PAT.FULL._1976-2010/FEB 11
       Examined 450 files
       Examined 500 files
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
               1   996: Newsroom 2004_

   5 files have one or more items; file list includes 508 files.
   One or more terms were invalid in 66 files.
```

? b hits

```
       24feb10 08:19:44 User276702 Session D274.2
          $133.68   43.263 DialUnits File411
   $133.68  Estimated cost File411
     $3.74  INTERNET
   $137.42  Estimated cost this search
   $137.56  Estimated total session cost  43.514 DialUnits

SYSTEM:OS  - DIALOG OneSearch
  File  15:ABI/Inform(R) 1971-2010/Feb 23
         (c) 2010 ProQuest Info&Learning
  File 324:GERMAN PATENTS FULLTEXT 1967-201003
```

```
         (c) 2010 UNIVENTIO/THOMSON
  File 349:PCT FULLTEXT 1979-2010/UB=20100205|UT=20100204
         (c) 2010 WIPO/Thomson
  File 654:US PAT.FULL. 1976-2010/FEB 11
         (c) Format only 2010 Dialog
*File 654: Reassignment data for 2009 delayed due to technical issues.
See File 123 for current reassignments.
  File 996:Newsroom 2004
         (c) 2009 Dialog

      Set  Items  Description
      ---  -----  -----------
```

? s (semiadditive? or (semi () additive)) (n2) (measure? or metric? or value? or parameter?) and (cube? or account? or microsoft) not py>2004

Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing

```
Processing
Processing
Processing
Processing
             261  SEMIADDITIVE?
         1041742  SEMI
          541676  ADDITIVE
            2127  SEMI(W)ADDITIVE
         5313270  MEASURE?
          321000  METRIC?
         6440695  VALUE?
         1984719  PARAMETER?
              13  (SEMIADDITIVE? OR SEMI(W)ADDITIVE)(2N)(((MEASURE? OR
                  METRIC?) OR VALUE?) OR PARAMETER?)
          137107  CUBE?
         3256166  ACCOUNT?
          487028  MICROSOFT
         6051580  PY>2004
      S1       6  (SEMIADDITIVE? OR (SEMI () ADDITIVE)) (N2) (MEASURE? OR
                  METRIC? OR VALUE? OR PARAMETER?) AND (CUBE? OR ACCOUNT?
                  OR MICROSOFT) NOT PY>2004
```

? rd

```
>>>Duplicate detection is not supported for File 324.

>>>Duplicate detection is not supported for File 349.
```

```
>>>Duplicate detection is not supported for File 654.

>>>Records from unsupported files will be retained in the RD set.
      S2       5  RD  (unique items)
```

? t s2/3,k/all

Dialog eLink: USPTO Full Text Retrieval Options

2/3,K/1 (Item 1 from file: 15)
DIALOG(R)File 15: ABI/Inform(R)

02743296 477856671

Applying UML and XML for designing and interchanging information for data warehouses and OLAP applications

Trujillo, Juan; Lujan-Mora, Sergio; Song, Il-Yeol
Journal of Database Management v15n1 pp: 41-72
Jan-Mar 2004
ISSN: 1063-8016 **Journal Code:** DAN
Word Count: 10371

Abstract:

...by the UML to simplify the final model. Furthermore, a UML-compliant class notation (called cube class) is provided to represent OLAP users' initial requirements. The paper also describes how to...

Text:

...UML to simplify the final model. Furthermore, we provide a UML-compliant class notation (called cube class) to represent OLAP users' initial requirements. We also describe how we can use the...

...we have provided a UML-compliant class notation to represent OLAP users' initial requirements (called cube class). From these cube classes, we then describe the use of state and interaction diagrams to model the behavior...The next section details the major features of MD modeling that should be taken into account for a proper MD conceptual design. Then the paper summarizes the most relevant conceptual approaches...

...fact attributes (atomic or derived), which are contained in cells or points in the data cube . This set of measures is based on a set of dimensions that determine the granularity...

...and time. On the left hand side of Figure 1, we can observe a data cube typically used for representing an MD model. In this particular case, we have defined a cube for analyzing measures along the product, store and time dimensions.

We note that a fact...

...OLAP terminology), however, might not be semantically meaningful for all measures along all dimensions. A measure is semi - additive if the SUM operator can be applied to some dimensions, but not all the dimensions...

...additive measures would be those measures that record a static level such as inventory, financial account balances or measures of intensity such as room temperatures (Kimball and Ross, 2002).

Figure 1: A data cube and classification hierarchies defined on dimensions

Regarding dimensions, the classification hierarchies defined on certain dimension...number of dimensions increases significantly, which may then lead to extremely sparse dimensions and data cubes . In this way, there are some attributes that are normally valid for all elements within... diagram. Secondly, we summarize how users' initial requirements are easily considered by what we call cube classes. We next describe how to model the behavior of MD databases by using UML state and interaction diagrams from the information represented in these cube classes. The next section sketches how we automatically transform an MD model accomplished by following...

...us to specify users' initial requirements by means of a UML-compliant class notation called cube class. After requirements are specified, behavioral properties are usually then related to these cube classes that represent users' initial requirements. We particularly use state and interaction diagrams to model the behavior (evolution) of these cube classes based on the applied OLAP operation.

Cube classes follow the query-by-example (QBE) method: the requirements are defined by means of...QBE systems are considered easier to learn than formal query languages. The structure of a cube class is shown in Figure 4:

* Cube class name.

* Measures area, which contains the measures from the fact to be analyzed.

* Slice...

...conditions to address the analysis.

* Order area, which specifies the order of the result set.

* Cube operations, which cover the OLAP operations for a further data-analysis phase.

We should point out that this graphical notation of the cube class aims at facilitating the definition of users' initial requirements to non-expert UML or databases users. In a more formal way, every one of these cube classes has its underlying OQL specification. Moreover, an expert user can directly define cube classes by specifying the OQL sentences (see more details on the representation of cube classes further in the paper).

Figure 4: Cube class structure

Behavioral Properties by Using State and Interaction Diagrams

From these cube classes seen in the previous section, final users may start a navigational process by applying...

...etc.) in the further data analysis phase. These operations are closed as

they generate another cube class as an output. Thus, we use state and interaction diagrams2 to model the behavior (evolution) of these cube classes based on the applied OLAP operation. These diagrams contain information about the most probable...

...proper view maintenance policy.

Regarding state diagrams, one state diagram is defined for each initial cube class. The diagram specifies that certain OLAP operations lead users to cube classes that allow them to analyze the same data (the same measures along the same...

...considered as a valid state. Every one of these valid states will be a new cube class. Then, the provided OLAP operations allow us to navigate along the states to define new cube classes.

In Figure 5, we can see an example of state diagrams. One state is...

...in the classification hierarchy of the dimensions included in the corresponding Dice area of the cube class. The data analysis will start on the initial state that corresponds to the finest...

...1998; OMG, 2001) for their clarity and low complexity. This interaction diagram shows interactions among cube classes, changed by OLAP operations such as rotate, pivot, slice, or dice. Thus, we can specify that certain OLAP operations (e.g., dice) lead users to cube classes which will show completely different data. Thus, these new cube classes represent the most probable new requirements a final user wishes to execute.

In Figure...

...see an example of interaction diagrams. Let us suppose that we have only defined two cube classes to specify two initial requirements. Then, we specify the operation needed to switch from one cube class into the other. In this particular case, the rotate operation indicates the transition that...

...an MD model in most of the relational database management systems such as Oracle, Informix, Microsoft SQL Server, IBM DB2 and so on (Trujillo et al., 2001b). Furthermore, we also provide...that cannot be directly represented by using this standard and that should be taken into account when transforming this ODBM into a particular object-oriented model of the target ODBMS.

The...

...used to define the specifications of object types. Then, we will sketch how to represent cube classes into the OQL, a query language that supports the ODMG data model. The great...should be used. Unfortunately, a constraint language is completely missing from the ODMG standard specification.

Cube Classes Represented by Using OQL

The OQL is not easy to use for defining users...

...underlying ODL representation corresponding to the MD model. Due to this fact, we also provide cube classes, which allow the user to define initial requirements in a graphical way. These cube classes can automatically be transformed into OQL sentences, and can therefore be used to query...

...ordered by the Brand of the product

In Figure 12, we can see the corresponding cube class to the previous requirement. It is easy to see how the cube class is formed:

* Measures contains the goal of the analysis: SUM(quantity).

* Slice the restrictions...

...the Product and Time dimensions.

* And Order defines the order of the result set.

The cube class can be automatically translated into OQL. The algorithm uses the corresponding ODL definition of...

...year in Month base class. Moreover, when attributes' names are omitted in the cube class, the algorithm automatically selects the descriptor attribute defined in the MD model. For example, the expression Time.Month= "January" of the cube class in Figure 12 involves the use of the descriptor attribute from the Month base...

...Brand involves the use of the descriptor attribute from Brand. The OQL for the corresponding cube class in Figure 12 is as follows:

Figure 12: An example of a user's...1) contains star schema packages (PKSCHEMAS) with dependencies between them (DEPENDENCIES) and users' initial requirements (CUBE CLASSES).

In this example, the MD model only contains one star schema package (Figure 7...

...is not any dependency between star schema packages, the DEPENDENCIES element is empty. Finally, the CUBECLASSES element is also empty as no initial requirement has been specified yet.

In our DTD...the left hand side of Figure 18, we can observe the graphical notation of the cube class that corresponds to this requirement. The measure to be analyzed (quantity...

...included in the dice area. Finally, the available OLAP operations are specified in the CO (Cube Operations) section (in this example the CO are omitted to avoid unnecessary detail). On the right hand side of Figure 18, the OQL sentence corresponding to the cube class is shown. We can notice how the descriptor attributes from the MD model are...

...levels are omitted in the analysis. For example, the expression Warehouse.State= "Valencia" of the cube class involves the use of the descriptor attribute from the State base class (Figure 16).

Regarding state diagrams, one state diagram is defined for each initial cube class. The diagram specifies that certain OLAP operations lead users to cube classes that allow them to analyze the same data (the same measures along the same...

...ways. For example, in Figure 19 we can see the corresponding state diagram of the cube class definition of Figure 18. It may be observed, for example, that roll-up and...

...can also be defined for each UML class diagram. This interaction diagram shows interactions among cube classes, changed by OLAP operations such as rotate, pivot, slice, or dice. In Figure 20, we can see an example of an

interaction diagram, in which we have considered three cube classes that specify the user's initial requirements. We have then defined the OLAP operations needed to switch between these cube classes.

Figure 18: An example of a user's initial requirement for the Warehouse case...
...a large bank is presented. The bank offers a significant portfolio of financial services: checking accounts , savings accounts , mortgage loans, safe deposit boxes, and so on.

This example introduces the following concepts:

* Heterogeneous...

...of their classification hierarchies.

Figure 21 represents level 1, which comprises five star packages: Saving Accounts Star, Personal Loans Star, Investment Loans Star, Safe Deposit Boxes Star, and Mortgage Loans Star...

...is shown in Figure 23. To avoid unnecessarily complicating the figure, three of the dimensions (Account , Time, and Status) with their corresponding hierarchies are not represented. Moreover, the attributes of the...of dimension attributes. Regarding dynamic aspects, we provide a UML-compliant class graphical notation (called cube classes) to specify users' initial requirements at the conceptual level. We have also described how...

...use state and interaction diagrams to model the behavioral aspects of the system regarding these cube classes based on the set of the applied OLAP operations. Moreover, we have sketched out...

Dialog eLink: Order File History
2/3,K/2 (Item 1 from file: 324)
DIALOG(R)File 324: GERMAN PATENTS FULLTEXT

0002623309
SCHALTUNGSTRAEGER FUER HOCHFREQUENZLEITUNGEN
CIRCUIT BEARERS FOR HIGH FREQUENCY MANAGEMENTS
Patent Applicant/Assignee:
STANDARD ELEKTRIK LORENZ AG 7000 STUTTGART, DE,, DE
Inventor(s):
FLORJANCIC MATJAZ, DR., 7141 MURR, DE,, DE
RICHTER HORST, DR., 7030 BOEBLINGEN, DE,, DE
HOLDIK KARL, DR.RER.NAT., 7000 STUTTGART, DE,, DE
RAUSCHMAIER RUEDIGER, DR.RER.NAT., 7441 ALTENRIET, DE,, DE

Publication & Filing Information

		Kind	Date

	Serial Number		
Publication	DE 3838486	A1	19900517
Application	DE 3838486		19881112

Priority application(s): DE 3838486 19881112 (Original format: DE 3838486)
Publication Language: German ; Application Language: German
Fulltext Word Count (English): 2882
Fulltext Word Count (German) : 2436
Fulltext Word Count (Both) : 5318 Fulltext Availability: Description (English machine translation)**Description** (English machine translation)...enclosure thereby both the full additive technology and the by a such structure take strong measures he semi - additive technology at the same time hand that the radiation can be kept negligibly small by... ...from the elek ten themselves 32 to a multi-long arrangement connected. In addition trotechnischen accounts receivable and defaults resulting in adhesive films, so-called prepregs 33.34 of zwi conductive... **Description** (German)

Dialog eLink: Order File History
2/3K/3 (Item 1 from file: 349)
DIALOG(R)File 349: PCT FULLTEXT

01140955

SPECIFYING MULTIDIMENSIONAL CALCULATIONS FOR A RELATIONAL OLAP ENGINE
SPECIFICATION DE CALCULS MULTIDIMENSIONNELS POUR MOTEUR DE TRAITEMENT ANALYTIQUE EN LIGNE RELATIONNEL

Patent Applicant/Patent Assignee:

- **INTERNATIONAL BUSINESS MACHINES CORPORATION**
  New Orchard Road, Armonk, NY 10504; US; US(Residence); US(Nationality)
- **IBM UNITED KINGDOM LIMITED**
  PO Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU; GB; GB(Residence); GB(Nationality); (Designated only for: MG)

Inventor(s):

- **COLOSSI Nathan Gevaerd**
  3621 Copperfield Drive, Apartment 342, San Jose, CA 95136; US
- **MALLOY William Earl**
  2637 Gamblin Drive, Santa Clara, CA 95051; US
- **PIRAHESH Mir Hamid**
  1282 Quail Creek Circle, San Jose, CA 95120; US

- **TOMLYN Craig Reginald**
  4501 Snell Avenue, #310, San Jose, CA 95136; US

Legal Representative:

- **WILLIAMS Julian David (agent)**
  IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN; GB

	Country	Number	Kind	Date
Patent	WO	200463942	A1	20040729

ApplicationWO2003GB549020031217

PrioritiesUS200334176320030113

Designated States: (Protection type is "Patent" unless otherwise stated - for applications prior to 2004)
AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG,
BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ,
DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB,
GD, GE, GH, GM, HR, HU, ID, IL, IN, IS,
JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS,
LT, LU, LV, MA, MD, MG, MK, MN, MW, MX,
MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO,
RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM,
TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU,
ZA, ZM, ZW

[EP] AT; BE; BG; CH; CY; CZ; DE; DK; EE; ES;
FI; FR; GB; GR; HU; IE; IT; LU; MC; NL;
PT; RO; SE; SI; SK; TR;

[OA] BF; BJ; CF; CG; CI; CM; GA; GN; GQ; GW;
ML; MR; NE; SN; TD; TG;

[AP] BW; GH; GM; KE; LS; MW; MZ; SD; SL; SZ;
TZ; UG; ZM; ZW;

[EA] AM; AZ; BY; KG; KZ; MD; RU; TJ; TM;

Publication Language:	English
Filing Language:	English
Fulltext word count:	23995

English Abstract:

...a system, method, and program for specifying multidimensional calculations. Selection of a subset of a cube model metadata object that

is generated from a facts metadata object and one or more... ...measure metadata objects. A statement is generated for retrieving multidimensional information using metadata in the cube model metadata object and the measure metadata objects, wherein each of the measure metadata objects...

French Abstract:

...a : recevoir une selection d'un sous-ensemble d'objet de metadonnees de modele de cube , produite a partir d'un objet de metadonnees de faits et d'un ou de... ...recuperer des donnees multidimensionnelles, a l'aide des metadonnees de l'objet de metadonnees modele cube et du ou des objet(s) de metadonnees de mesure, chacun de ces objets specifiant...

Detailed Description:

...nature of OLAP systems, the collections of data that they implement are referred to as cubes .: As for information, OLAP systems store and calculate information. Data for OLAP systems often come...
...of OLAP systems
in which special-purpose file systems or indexes are used to store cube data. Express Web Publisher, EssbaseTm, TMI, and Pilot Suite are a few examples of products based on special-purpose storage and indexing technology. Microsoft 's OLAP offering also includes a MOLAP engine.

These systems are often read-only systems... ...added to the SQL standard to
allow a single query to generate multiple aggregates: ROLLUP, CUBE , and GROUPING SETS. These super aggregate operators are extensions to the GROUP BY clause and... ...on-line analytical processing (OLAP) systems (e.g.,
from companies such as Hyperion, Cognos, and Microsoft) are designed to return multidimensional result sets naturally, when given sets of members for each edge of a multidimensional cube . The multidimensional OLAP systems are also designed to compute some or all of the results...
...aspect, a
method for specifying multidimensional calculations, comprising.

receiving selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more ... ...measure metadata objects; and generating a statement for
retrieving multidimensional information using metadata in the cube model metadata object and the one or more measure metadata objects, wherein each of the... ...statement
further comprises: generating a SELECT statement for a slice of the subset of the cube model metadata object.

Preferably, generation of the structured query language statement further comprises: generating a SELECT statement for the subset of the cube model metadata object.

Preferably, the method further comprises: separating symmetric measures and asymmetric measures defined... ...the measures may be rewritten,
rewriting the measures.

Preferably, the method further comprises: generating a cube metadata object based on the selection of the subset of the cube model metadata object, including generating a structured query language statement for creation of a cube view, wherein the structured query language statement is generated from metadata in the one or... ...metadata objects.

Preferably, the method further comprises: under control of an application program, using the cube model metadata object and one or more of the measure metadata objects, generating a structured... ...component to operate at least one
program for: receiving selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more... ...measure metadata objects; and generating a statement for
retrieving multidimensional information using metadata in the cube model metadata object and the one or more measure metadata objects, wherein each of the... ...at least one program further comprises: generating a SELECT statement for the subset of the cube model metadata object.

Preferably, the at least one program further comprises: generating a SELECT statement for the subset of the cube model metadata object.

Preferably, the at least one program further comprises: separating symmetric measures and ... ...be rewritten, rewriting the measures.

Preferably, the at least one program further comprises: generating a cube metadata object based on the selection of the subset of the cube model metadata object, including generating a structured query language statement for creation of a cube view, wherein the structured query language statement is generated from metadata in the one or... ...the at least one program further comprises: under
control of an application program, using the cube model metadata object and one or more of the measure metadata objects, generating a structured... ...causes operations to be performed, the operations comprising.

receiving selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more... ...measure metadata objects; and generating a statement for
retrieving multidimensional information using metadata in the cube model metadata object and the one or more measure metadata objects, wherein each of the... ...statement further comprise: generating a SELECT statement for a
slice of the subset of the cube model metadata object.

Preferably, the operations for generation of the structured query

language statement further comprise: genera-ting a SELECT statement for the subset of the cube model metadata object.

Preferably, the operations further-comprise: separating symmetric measures and asymmetric measures defined... ...the measures may be rewritten, rewriting the measures.

Preferably, the operations further comprise: generating a cube metadata object based on the selection of the subset of the cube model metadata object, including generating a structured query language statement for creation of a cube view, wherein the structured query language statement is generated from metadata in the one or...
...metadata objects.

Preferably, the operations further comprise: under control of an application program, using the cube model metadata object and one or more of the measure metadata objects, generating a structured... ...a method, system, and program for
specifying multidimensional calculations. Selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more... ...measure metadata objects. A statement is
generated for retrieving multidimensional information using metadata in the cube model metadata object and the measure metadata objects, wherein each of the measure metadata objects... ...certain implementations of the invention.

FIG. 5 illustrates that metadata objects fit together in a cube model and map to a relational star schema of relational tables in accordance with certain... ...of the invention.

FIG. 10 illustrates instances of met&data objects used to define a cube in accordance with certain implementations of the invention.

FIG. 11 illustrates that one instance of... ...objects in accordance with certain implementations of the invention.

FIG. 23 illustrates creation of a semi - additive measure in accordance with certain implementations of the invention.

FIG. 24 illustrates creation of a composite... ...can run on computers using various operating systems, such as IBM z/OS@, IBM AIX@, Microsoft Windows@ 2000, Microsoft Windows@ XP, Linux, Solaris, HP- UX.

FIG. 1 illustrates, in a block diagram, a computing... ...be grouped together by their relationships to each other, into
a metadata object called a cube model. A cube model represents a particular grouping and configuration of relational tables. The purpose of a cube model is to describe OLAP structures to a given application or tool. Cube models tend to describe all cubes that different users might want for the data that are being analyzed. A cube model groups dimensions and facts, and offers the flexibility of for

dimensions. A cube model conveys the structural information needed by query design tools and applications that generate complex... ...whether or not the metadata
objects are included in a more complex multidimensional structure.

A cube model can be constructed in many ways, but is often built to represent a relational star schema or snowflake schema. A cube model based on a simple star schema is built around a central facts metadata object... ...of the invention.

Dimension metadata objects are connected to the facts metadata object in a cube model just as the dimension tables are connected to the fact table in a star... ...the State and City attributes, and also references the CityPop AR attribute relationship. In a cube model, each dimension can have multiple hierarchies, but the example star schema has one hierarchy... ...are
connected in a star shape to a central facts metadata object to create a cube model. Join metadata objects can join tables to create a facts metadata object or a dimension metadata object. Metadata joins can also act as glue within the cube model by joining facts metadata objects to dimension metadata objects. The dimension metadata objects have... ...attributes, and related joins.

FIG. 5 illustrates that metadata objects 500 fit together in a cube model and map to a relational star schema of relational tables 550 in accordance with certain implementations of the invention. A cube model metadata object 510 is built on dimension metadata objects 512, 514, join metadata objects 516, 518, and a facts metadata object 520.

Cube model metadata objects are flexible metadata objects whose components may be reused to create more precise cube metadata objects for specific applications. For example, a cube model metadata object may have 37 facts, but one cube metadata object generated from the cube model metadata object may eliminate one or more dimension metadata objects, one or more levels... ...a dimension metadata object, and/or one or more measure metadata objects.

in addition to cube model metadata objects, there is a more specific metadata object called a cube metadata object. A cube metadata object is the closest metadata object to an OLAP conceptual cube . A cube metadata object is a specific instance or subset of a cube model metadata object.

A cube metadata object has a specific set of similar but more restrictive metadata objects derived from the parent cube model metadata object including: cube dimensions, cube hierarchies, and cube facts. For example, a RegionCubeDim is a cube dimension that is a subset of attributes derived from the Region dimension. RegionCubeDim references the... ...it scopes and all of the structural
information, including the join information, stays with the cube model Region dimension.

In certain implementations, a cube metadata object has one cube hierarchy defined per cube dimension, while a dimension metadata object can have many hierarchies defined for-the cube model metadata object.

This structural difference between a cube metadata object and a cube model
metadata object allows retrieval of a cube metadata object with a single SQL statement.

FIG. 6 illustrates that conceptual metadata objects are... ...OLAP structures. By grouping metadata objects from
other layers, the OLAP layer 620 provides-OLAP cubes with different degrees of complexity.

An example is provided for better understanding of the embodiment... ...relationship is referenced. All attribute
relationships that apply to a given hierarchy are captured. One cube hierarchy 850, 860, 870 per hierarchy is also created in order to be used in a cube context. The cube hierarchies 850, 860, 870 are used to scope the levels of a hierarchy that are interesting for a given cube . A cube hierarchy 850, 860, 870 also captures attribute relationships that apply to it.

FIG. 9 illustrates... ...FIG. 10 illustrates instances of metadata objects 1000, 1010, 1020,
1030 used to define a cube in accordance with certain implementations of the invention. A cube facts, cube dimension, and cube hierarchy metadata objects are used to scope the attributes and measures that are part of a cube . Each of these metadata objects references the metadata object that is being scoped, and all... ...information, such as joins, is kept in the main (i.e., parent) metadata object. All cube specific objects hold a reference to a main object from which they were defined. For example, the cube hierarchy metadata object has a reference to the hierarchy metadata object from which the cube hierarchy metadata object was defined. In certain implementations, for cube dimensions, one hierarchy is assigned.

In the example, a cube fact SalesCubeFacts 1000 is created and lists the measure (Sales) that is used in the cube .

The OLAP layer is composed by cube model and cube metadata objects.

A cube model metadata object describes the facts and dimensions that are interesting to a given application. The dimensions of a cube model metadata object can have multiple hierarchies defined, which makes a cube model metadata object a very flexible structure. A cube metadata object is derived from a cube model metadata object, and so all cube dimensions, cube hierarchies, and cube facts metadata objects are derived from the cube model metadata object. A difference between a cube model metadata object and a cube metadata object is that in a cube metadata object one hierarchy is defined per dimension, which makes it possible to retrieve a cube metadata object with a single SQL statement.

FIG. 11 illustrates that one instance of each... ...in an
OLAP layer is created in accordance with certain implementations of the invention. The cube model created in the example captures one possible cube model 1100 generated from the example star-join schema of FIG. 3. A cube 1150 is created based on the cube dimensions TimeCubeDim, ProductCubeDim, RegionCubeDim and cube facts SalesCubeFacts.

A.2 Metadata Oblect Pro-Perties
Each metadata object has a set of... ...metadata object specific
properties describe
the components and qualities that define the metadata object.

The cube model is a representation of a logical star schema. The cube model is a grouping of relevant dimension metadata objects around a central facts metadata object. Each dimension can have multiple hierarchies, which increases the flexibility of the cube model. The structural information about how to join the tables used by the facts and dimension metadata objects is stored in the cube model. Also stored in the cube model is enough information to retrieve OLAP data. Other reporting and OLAP tools that understand the cube model and can handle multiple hierarchies of a specific dimension can benefit from the use of a cube model.

Cube models define a complex set of relationships and can be used to selectively expose relevant... ...central facts metadata
object is stored with the corresponding dimension as a set. Subsets of cube model components can be used by many cubes for different analysis purpo@es.

An empty cube model may be created that does not have a facts metadata object or any dimensions. However, the cube model is completed before creating a corresponding cube . The OLAP multidimensional metadata system 100 validates a cube model by ens@uring that the cube model includes
a facts metadata object, at least one dimension, and joins between the existing... ...all of the attributes reference
valid tables. A hierarchy is not required to consider a cube model complete, however, to be able to define a cube from a cube model, at least one hierarchy per dimension is defined.

Each metadata object has a set... ...components and qualities that define the
metadata object. The metadata object specific properties of a cube model are described Table 2 in accordance with certain implementations of the invention.

Table 2
Property Description
Facts Facts used in the cube model.

Set of Dimensions that are used in the cube model
(dimension and their corresponding joins.

. join)
The facts metadata object groups related measure$ which... ...of attributes and a set of joins. A facts metadata object is used in a cube model as the center of a star schema.

The facts metadata object plays the role... ...of related attributes that together describe one aspect of a measure. Dimensions are used in cube models to organize the data in the facts metadata object according to
logical categories such... ...relationships among a set of one or more attributes within a given dimension of a cube model. Defining these relationships provides a navigational and computational means of traversing a given dimension. Multiple hierarchies can be defined for a dimension of a cube model. The hierarchy metadata object also references a set of attribute relationships that link attributes... ...each measure, a list of
aggregations is defined for calculations in the context of a cube model, or cube . Each aggregation in the list specifies an aggregation function, such as SUM, COUNT, MIN, MAX... ...the dimension set is empty, the aggregation function is applied to all dimensions in the cube or cube model that are not specifically being used by another aggregation in the list. In certain... ...example of a simple measure is Revenue. The Revenue measure can
be created for a cube model with three dimensions: Product, Market and Time. Revenue has a SQL expression template (template... two dimensions, or a fact and a dimension. Join metadata objects are referred to by cube model, facts, and dimension objects.

The metadata object specific properties that define a join metadata... ...Cardinality Cardinality expected in the join.

(1:1, 1:N,
N:1, N:N]
A cube is a very precise definition of an OLAP cube that can be delivered using a single SQL statement. Each cube is derived from a single cube model. The cube facts and list of cube dimensions are subsets of those in the referenced cube model. A cube view name is also defined which represents the cube in the database. Cubes are appropriate for tools and applications that do not use multiple hierarchies because cube dimensions allow one cube hierarchy per cube dimension.

The purpose of a cube is to define a standard relational view of an OLAP structure. in addition to the relational view, a cube provides an extended describe (e.g., XML document) that describes the roles of its columns in multidimensional terms. In the process of defining a cube , the
designer selects a subset of the possible elements, choosing a single hierarchy for each dimension. This ensures that the cube unambiguously

defines a single relational result set. The simplicity of a cube makes the cube useful to less sophisticated OLAP applications, such as portable devices powered by World Wide Web ("Web") services.

The metadata object specific properties of a cube metadata object are described in the following Table 12 in accordance with certain implementations of the invention.

Table 12
Property Description
Cube model Cube model from which the cube is derived.

Cube facts Cube facts used.,in the cube . The cube
facts is derived from the facts metadata
obj ect in the cube model.

List of Ordered list of cube dimensions used in the
cube cube . The cube dimension is derived from
dimensions the dimensions in the cube model. One cube
hierarchy is associa
@ted with each cube
dimension.

Cube view View in the database that represents the
cube .

Extended XML document describing roles of columns
Describe and their relationships in terms of a
multidimensional model
A cube facts metadata object has a subset of measures in an ordered list from a specific facts metadata object. A cube facts metadata object gives a cube the flexibility to scope facts for a cube model. The structural information, such as the joins and attributes, is referenced from the parent facts metadata object. The metadata object specific properties that define a cube facts metadata object are described in the following Table 13 in accordance with certain implementations of the invention.

- Table 13
Property Description
Facts Facts from which the cube facts is derived.

List of Ordered list of measures used in a cube .

measures All measures are part'of the facts from
which the cube facts is derived.

A cube dimension metadata object is used to scope a dimension for use in a cube . The cube dimension metadata object references the dimension from which it is derived and the relevant cube hierarchy for the given cube . In certain implementations, one cube hierarchy can be

applied
to a cube dimension. The joins and attributes that apply to the cube dimension are referenced from the dimenslon definition. The metadata object specific properties that define a cube dimension metadata object are described in the following Table 14 in accordance with certain implementations of the invention.

Table 14

Property	Description
Dimension	Dimension from which the cube dimension is derived.
Cube hierarchy	Cube hierarchy that applies to the cube dimension.

A cube hierarchy metadata object is a scoped version of a hierarchy and is used in a cube . A cube hierarchy references the hierarchy from which it is derived and can have a subset of the attributes from the parent hierarchy. Additionally, a cube hierarchy metadata object references the attribute relationships that apply on the cube . In certain implementations, one cube hierarchy can be defined for a cube dimension of a cube . A cube hierarchy metadata object has the same hierarchy types and
deployment mechanisms as the hierarchy from which the cube hierarchy metadata object is derived.

A cube hierarchy is very similar to a hierarchy; however, a cube dimension refers to a single cube hierarchy. This allows a single SELECT statement to calculate the cells of a cube .

The metadata object specific properties that define a cube hierarchy metadata object are described in the following Table 15 in accordance with certain implementations of the invention.

Table 15

Property	Description
Hierarchy	Hierarchy from which the cube hierarchy is derived.
Li sts of attributes	Ordered list of all attributes from the top to the bottom of the cube hierarchy. The order of the attributes should be the same as in the parent hierarchy.
Set of attribute relationsh ip s	Set of all attribute relationships that link cube hierarchy-attributes to other attributes.

FIG. 16 illustrates some relationships among some... ...invention. The arrows

indicate that a metadata object references another metadata object. For example, a cube metadata object 1610 references a cube model metadata object 1600. A more detailed relationship description of the metadata objects is illustrated... ...implementations of the invention.

Table 16
Netadata References Metadata
Metadata object Metadata object 2
1
Cube Model zero or one Facts
Cube Model zero or more Dimension/join
Cube one Cube model
Cube one Cube Facts
Cube one or more Cube Dimension
Facts one or more Measure
Facts zero or more Attribute
Facts zero or more... ...Dimension one or more Attribute
Dimension zero or more Join
Dimension zero or more Hierarchy
Cube Facts one Facts
Cube Facts one or more Measure
Cube Dimension one Dimension
Cube Dimension one or more Attribute
Cube Dimension one Cube Hierarchy
Hierarchy one or more Attribute
Hierarchy zero or more Attribute
Relationship
Cube Hierarchy one Hierarchy
Cube Hierarchy one or more Attribute
Cube Hierarchy zero or more Attribute
Relationship
Measure zero or more Measure
Measure zero or more... ...object rules for that
metadata object. The rules are separated in three categories: Base Rules, Cube Model Completeness Rules, and Optimization Rules. The following discussion of specific rules provides a set... ...be modified without departing
from the scope of the invention.

The base rules for a cube model metadata object are: (1) the cube model metadata object refers to zero or one facts metadata object; (2) the
cube model metadata object refers to zero or more dimension(s); (3) dimension-join pairs have... ...the dimension
metadata object; and (5) for each measure referenced in the facts of the cube model, all the explicit dimension references in the aggregations of the measure are referenced by the cube model. When the cube model references at least one dimension, an aggregation with an empty dimension set matches to at least one dimension from the cube model that was not previously referenced.

The base rules for a cube metadata object are: (1) the cube metadata object refers to one cube facts; (2) the qjIbe metadata object refers to at least one cube dimension; (3) cube facts is derived from the facts used in
the cube model; and, (4) cube dimensions are derived from the dimensions used in the cube model.

The base rules for a facts metadata object are: (1) a facts metadata object... ...by a dimension refer to the attributes of the dimension.

The base rules for a cube facts metadata object are: (1) the cube facts metadata object refers to at least one facts; (2) the cube facts metadata object refers to at least one measure; and, (3) measures referenced by a cube facts metadata object are part of the facts metadata object.

The base rules for a cube dimension metadata object are as follows.

(1) the cube dimension metadata object refers to one dimension; (2) the cube dimension metadata object refers to a cube hierarchy; and, (3) the cube hierarchy referenced by the cube dimension metadata object is derived from a hierarchy that is referenced by the dimension of the cube dimension metadata object.

The base rules for a hierarchy metadata object are: (1) the hierarchy... ...Deployment
Balanced X
Ragged X
Unbalanced X X
Network X
The base rules for a cube hierarchy metadata object are: (1) the cube hierarchy metadata object refers to one hierarchy; (2) the cube hierarchy metadata object refers to at least one attribute; (3) attributes
referenced by the cube hierarchy metadata object are part of the hierarchy; (4) the order of the attributes in the cube hierarchy metadata object are the same as in the hierarchy (with the exception of hierarchies... ...a left attribute as part of the hierarchy; and, (6) attribute
relationships referenced in the cube hierarchy metadata object are also referenced in the hierarchy that defines the cube hierarchy.

The base rules for a measure metadata object are: (1) a measure metadata object... ...left and right attributes, as well as the operation defined for them, are compatible.

The cube model completeness rules extend the base rules in order to ensure that a cube model has the required links to other metadata objects to allow effective warehouse SQL queries to be formed. The cube model..

completeness rules for a cube model metadata object are: (1) a cube model metadata object refers to one facts; (2) a cube model metadata object refers to one or more dimensions.

The optimization rules extend the cube model completeness rules in order to ensure that optimization of warelf6use SQL queries can be performed.

The optimization rules for a cube model metadata object is: (1) the join used in the facts to dimension has a... ...properties in FIGs. 18A-18E are common properties. For example, FIGs.

18A- 18E illustrate a cube model metadata object instance 1800, a cube metadata object instance 1802, a facts metadata object instance 1804, a cube facts metadata object instance 1806, measure metadata object instances 1808, 1810, dimension metadata object instances 1812, 1814, cube dimension metadata object instances 1816, 1818, hierarchy metadata object instances 1820, 1822, 1824, cube hierarchy metadata object instances 1826,
1828, join metadata object instances 1830, 1832, and attribute metadata... ...user may use the user interface 150 to create metadata objects.

After creating an empty cube model metadata object, a facts metadata object and dimension metadata objects are created and joined to the cube model metadata object by creating appropriate join metadata objects.

The properties of the metadata objects... ...the dimension set is empty, the aggregation function is applied to all dimensions in the cube or cube model that are not specifically being used by any other aggregations in the list.

The multidimensional metadata software 120 automatically generates a SQL statement for generation of a cube view using the metadata in the measure metadata object.

B.1 Re irements for Measures...a measure. The calculation order for the set of measure
metadata objects referenced by a cube model metadata object or a cube metadata object need not be the same eacl@ measure metadata object may specify a calculation... ...independent expressions for each aggregation function input.

Yet another requirement for measures is support for semi - additive measures , such as snapshot measures (e.g., inventory). For example, for Market and Product dimensions, a... ...Product), (AVG, Time), (MAX, Market)). Targeting to sophisticated
applications is a more generic representation of semi - additive measures
,
which are described further below with reference to FIG. 23. Also, the requirement of targeting... ...the aggregation function of SUM.

The Cost measure metadata object 2210 is created for a cube model with the three dimensions: Product 22G2, Market 2204, and Time 2206. The Cost measure... ...the SQL: SUM(Fact.Cost).

The Revenue measure metadata object 2220 is created for a cube model with three dimensions: Product 2202, Market 2204, and Time 2206. The Revenue measure metadata... ...aggregation function., resulting in the SQL: SUM(Fact.Rev).

FIG. 23 illustrates creation of a semi - additive measure in accordance with certain implementations of the invention. The Inventory measure metadata object 2310 has... ...Each hierarchy metadata object includes information used to build a ROLLUP clause. In particular, a cube model metadata object may reference several possible hierarchies, but selection of a single hierarchy is required for SQL generation, in certain implementations of the invention. in block 2904, a cube model description of a cube model metadata object that references the facts and one or more dimension metadata objects is received, and the cube model metadata object is generated from the cube model description. In block 2906, a selection of a subset of the cube model metadata object is received. In block 2908, a cube metadata object is generated based on the selection. Additionally, a SQL statement is generated for creating a cube view from metadata in one or more measure metadata objects. The generation of the SQL... ...in dimension metadata objects). A ROLLUP for each
dimension returns results that represent an OLAP cube , in a relational way. The combination of more than one ROLLUP operator in a single... ...the following grouping clauses, which are a set of grouping clauses that make up a cube .

(country, state, year, month)
(country, state, year)
(country, state)
(country, year, month)
(country, year)
(country ... ...and state columns.

Although FIG. 29A describes use of measure metadata objects to generate a cube view, in additional implementations, the measure metadata objects may be used without creating a cube view. That is, the cube view' is one way to use the definitions of the measure metadata objects in order... ...the
measure metadata objects is for an application to read the measure definitions and the cube model metadata and to directly generate a SQL statement from the measure definitions and cube model metadata.

FIG. 29B illustratesdogic for generating a SQL statement from one or more measure metadata objects and a cube model metadata object in accordance with certain implementations of the invention. Control begins in block... ...Each hierarchy metadata

object includes information used to build a ROLLUP clause. In particular, a cube model metadata object may reference several possible hierarchies, but selection of a single hierarchy is required for SQL generation, in certain implementations of the invention. In block 2914, a cube model description of a cube model metadata object that references the facts and one or more dimension metadata objects is received, and the cube model metadata object is generated from the cube model description. in block 2916, a selection of a subset of the cube model metadata object is made by
an application program. In block 2918, under control of the application program, using the cube model metadata object and one or more of the measure metadata objects, a SQL statement... ...120 handles various query types (e.g.,
Grand Total query, Slice based query, and complete cube query). In a Grand Total slice, only the grand total for all dimensions is returned. A slice is a sub- cube , while a complete cube is an entire cube .

.Measures are represented as measure metadata objects. When multiple measures are to be calculated in... ...have little or no symmetry) or specify conflicting calculation
order for the dimensions of the cube , then those asymmetric measures are divided into subsets sharing the same calculation order and a... ...that retrieves multidimensional information.

In certain implementations, the SQL statement may be catalogued as a cube view and referenced by a cube metadata object.

in terms of calculating multiple measures, symmetry of a measure, distributiveness of aggregation...in a
report. However, the result of executing the SQL statement generated for a complete cube query is a cube view, which may itself be queried.

For all types of queries, the generation of the... ...f.timeid = t.timeid
group by m.Country, m.State, t.Year
For a complete cube query with symmetric measures, the multidimensional metadata software 120 may generate the following select statement... ...in a
report. However, the result of executing the SQL statement generated for a complete cube query is a cube view, which may itself be queried.

A set of asymmetric measures is calculated by using ... ...State, t.Year)
s
group by s.Country, s.State, s.Year
For a complete cube query with asymmetric measures, the multidimensional metadata software 120 may generate the following select statement... ...same set of attributes when these SQL
statements are generated for the same slice or cube . Therefore, the attribute instances will be the same in all the SQL statements. The technique... ...depends on the type of SQL statements being combined (i.e.,
slice-based vs. complete cube). For slice-based SQL statements, the

clause used in the inner join will use a... ...1 as GrandTotal
from Fact f) r2
ON rl.GrandTotal = r2.GrandTotal
For the complete cube type of SQL statement, the join clause also takes into consideration the fact that the... ...joins attribute instances when they contain specific members or contain aggregation. The following is a cube - based SQL statement generated by the multidimensional metadata software 120 for the first option that...
...Machines Corporation in the United States and/or other countries. Windows is a trademark of Microsoft Corporation in the United States and/or other countries. Solaris and JDBC are trade-marks...

Claims:

1 A method for specifying multidimensional calculations, comprising:receiving selection of a subset of a cube model metadata object that isgenerated from a facts metadata. object and one or more...
...measure metadata objects; andgenerating a statement for retrieving multidimensionalinformation using metadata in the cube model metadata object and theone or more measure metadata objects, wherein each of the... ...componentto operate at least one program for:receiving selection of a subset of a cube model metadata objectthat is generated from a facts metadata object and one or more... ...measure metadata objects; andgenerating a statement for retrieving multidimensionalinformation using metadata in the cube model metadata object and theone or more measure metadata objects, wherein each of the...

Dialog eLink: Order File History
2/3,K/4 (Item 1 from file: 654)
DIALOG(R)File 654: US PAT.FULL.

0005724064
Derwent Accession: 2004-570153
Method, system, and program for specifying multidimensional calculations for a relational OLAP engine
Inventor: Colossi, Nathan, INV
Malloy, William, INV
Pirahesh, Mir, INV
Tomlyn, Craig, INV
Assignee: International Business Machines Corporation 02)
Correspondence Address: KONRAD RAYNES VICTOR & MANN LLP, Suite 210 315 S. Beverly Drive, Beverly Hills, CA, 90212, US

Publication Number	Kind	Date	Application Number	Filing Date

Main Patent US 20040139061 A1 20040715 US 2003341763 20030113

Fulltext Word Count: 25660

**IMAGE Available

Abstract:
...a system, method, and program for specifying multidimensional calculations. Selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more...

...measure metadata objects. A statement is generated for retrieving multidimensional information using metadata in the cube model metadata object and the measure metadata objects, wherein each of the measure metadata objects...

Summary of the Invention:
...nature of OLAP systems, the collections of data that they implement are referred to as cubes . As for information, OLAP systems store and calculate information. Data for OLAP systems often come...

...of OLAP systems in which special-purpose file systems or indexes are used to store cube data. Express Web Publisher, Essbase(TM), TM1, and Pilot Suite are a few examples of products based on special-purpose storage and indexing technology. Microsoft 's OLAP offering also includes a MOLAP engine. These systems are often read-only systems...

...added to the SQL standard to allow a single query to generate multiple aggregates: ROLLUP, CUBE , and GROUPING SETS. These super aggregate operators are extensions to the GROUP BY clause and...

...on-line analytical processing (OLAP) systems (e.g., from companies such as Hyperion, Cognos, and Microsoft) are designed to return multidimensional result sets naturally, when given sets of members for each edge of a multidimensional cube . The multidimensional OLAP systems are also designed to compute some or all of the results...

...a method, system, and program for specifying multidimensional calculations. Selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more...

...measure metadata objects. A statement is generated for retrieving multidimensional information using metadata in the cube model metadata object and the measure metadata objects, wherein each of the measure metadata objects...

Description of the Drawings:
...0028]FIG. 5 illustrates that metadata objects fit together in a cube model and map to a relational star schema of relational tables in accordance with certain...

...0033]FIG. 10 illustrates instances of metadata objects used to define a cube in accordance with certain implementations of the invention...

...0046]FIG. 23 illustrates creation of a semi - additive measure in accordance with certain implementations of the invention...

Description of the Invention:
...on computers using various operating systems, such as IBM z/OS(R), IBM AIX(R), Microsoft Windows(R) 2000, Microsoft Windows(R) XP,

Linux, Solaris, HP-UX...

...be grouped together by their relationships to each other, into a metadata object called a cube model. A cube model represents a particular grouping and configuration of relational tables. The purpose of a cube model is to describe OLAP structures to a given application or tool. Cube models tend to describe all cubes that different users might want for the data that are being analyzed. A cube model groups dimensions and facts, and offers the flexibility of multiple hierarchies for dimensions. A cube model conveys the structural information needed by query design tools and applications that generate complex...

...0073] A cube model can be constructed in many ways, but is often built to represent a relational star schema or snowflake schema. A cube model based on a simple star schema is built around a central facts metadata object...

...0074] Dimension metadata objects are connected to the facts metadata object in a cube model just as the dimension tables are connected to the fact table in a star...

...the State and City attributes, and also references the CityPop AR attribute relationship. In a cube model, each dimension can have multiple hierarchies, but the example star schema has one hierarchy...

...are connected in a star shape to a central facts metadata object to create a cube model. Join metadata objects can join tables to create a facts metadata object or a dimension metadata object. Metadata joins can also act as glue within the cube model by joining facts metadata objects to dimension metadata objects. The dimension metadata objects have...

...0080]FIG. 5 illustrates that metadata objects 500 fit together in a cube model and map to a relational star schema of relational tables 550 in accordance with certain implementations of the invention. A cube model metadata object 510 is built on dimension metadata objects 512, 514, join metadata objects...

...0081] Cube model metadata objects are flexible metadata objects whose components may be reused to create more precise cube metadata objects for specific applications. For example, a cube model metadata object may have 37 facts, but one cube metadata object generated from the cube model metadata object may eliminate one or more dimension metadata objects, one or more levels...

...0082] In addition to cube model metadata objects, there is a more specific metadata object called a cube metadata object. A cube metadata object is the closest metadata object to an OLAP conceptual cube . A cube metadata object is a specific instance or subset of a cube model metadata object. A cube metadata object has a specific set of similar but more restrictive metadata objects derived from the parent cube model metadata object including: cube dimensions, cube hierarchies, and cube facts. For example, a RegionCubeDim is a cube dimension that is a subset of attributes derived from the Region dimension. RegionCubeDim references the...

...it scopes and all of the structural information, including the join information, stays with the cube model Region dimension...

...0083] In certain implementations, a cube metadata object has one cube hierarchy defined per cube dimension, while a dimension metadata object can have many hierarchies defined for the cube model metadata object. This structural difference between a cube metadata object and a cube model metadata object allows retrieval of a cube metadata object

with a single SQL statement...

...OLAP structures. By grouping metadata objects from other layers, the OLAP layer 620 provides OLAP cubes with different degrees of complexity ...

...relationship is referenced. All attribute relationships that apply to a given hierarchy are captured. One cube hierarchy 850, 860, 870 per hierarchy is also created in order to be used in a cube context. The cube hierarchies 850, 860, 870 are used to scope the levels of a hierarchy that are interesting for a given cube . A cube hierarchy 850, 860, 870 also captures attribute relationships that apply to it...

...FIG. 10 illustrates instances of metadata objects 1000, 1010, 1020, 1030 used to define a cube in accordance with certain implementations of the invention. A cube facts, cube dimension, and cube hierarchy metadata objects are used to scope the attributes and measures that are part of a cube . Each of these metadata objects references the metadata object that is being scoped, and all...

...information, such as joins, is kept in the main (i.e., parent) metadata object. All cube specific objects hold a reference to a main object from which they were defined. For example, the cube hierarchy metadata object has a reference to the hierarchy metadata object from which the cube hierarchy metadata object was defined. In certain implementations, for cube dimensions, one hierarchy is assigned. In the example, a cube fact SalesCubeFacts 1000 is created and lists the measure (Sales) that is used in the cube .

[...

...0091] The OLAP layer is composed by cube model and cube metadata objects. A cube model metadata object describes the facts and dimensions that are interesting to a given application. The dimensions of a cube model metadata object can have multiple hierarchies defined, which makes a cube model metadata object a very flexible structure. A cube metadata object is derived from a cube model metadata object, and so all cube dimensions, cube hierarchies, and cube facts metadata objects are derived from the cube model metadata object. A difference between a cube model metadata object and a cube metadata object is that in a cube metadata object one hierarchy is defined per dimension, which makes it possible to retrieve a cube metadata object with a single SQL statement...

...in an OLAP layer is created in accordance with certain implementations of the invention. The cube model created in the example captures one possible cube model 1100 generated from the example star-join schema of FIG. 3. A cube 1150 is created based on the cube dimensions TimeCubeDim, ProductCubeDim, RegionCubeDim and cube facts SalesCubeFacts...

...0097] The cube model is a representation of a logical star schema. The cube model is a grouping of relevant dimension metadata objects around a central facts metadata object. Each dimension can have multiple hierarchies, which increases the flexibility of the cube model. The structural information about how to join the tables used by the facts and dimension metadata objects is stored in the cube model. Also stored in the cube model is enough information to retrieve OLAP data. Other reporting and OLAP tools that understand the cube model and can handle multiple hierarchies of a specific dimension can benefit from the use of a cube model...

...0098] Cube models define a complex set of relationships and can be used to selectively expose relevant...

...central facts metadata object is stored with the corresponding dimension as a set. Subsets of cube model components can be used by many cubes for different analysis purposes...

...0099] An empty cube model may be created that does not have a facts metadata object or any dimensions. However, the cube model is completed before creating a corresponding cube . The OLAP multidimensional metadata system 100 validates a cube model by ensuring that the cube model includes a facts metadata object, at least one dimension, and joins between the existing...

...all of the attributes reference valid tables. A hierarchy is not required to consider a cube model complete, however, to be able to define a cube from a cube model, at least one hierarchy per dimension is defined...

...components and qualities that define the metadata object. The metadata object specific properties of a cube model are described Table 2 in accordance with certain implementations of the invention...

...of attributes and a set of joins. A facts metadata object is used in a cube model as the center of a star schema...

...of related attributes that together describe one aspect of a measure. Dimensions are used in cube models to organize the data in the facts metadata object according to logical categories such...

...relationships among a set of one or more attributes within a given dimension of a cube model. Defining these relationships provides a navigational and computational means of traversing a given dimension. Multiple hierarchies can be defined for a dimension of a cube model. The hierarchy metadata object also references a set of attribute relationships that link attributes...each measure, a list of aggregations is defined for calculations in the context of a cube model, or cube . Each aggregation in the list specifies an aggregation function, such as SUM, COUNT, MIN, MAX...

...the dimension set is empty, the aggregation function is applied to all dimensions in the cube or cube model that are not specifically being used by another aggregation in the list. In certain...

...example of a simple measure is Revenue. The Revenue measure can be created for a cube model with three dimensions: Product, Market and Time. Revenue has a SQL expression template (template...

...two dimensions, or a fact and a dimension. Join metadata objects are referred to by cube model, facts, and dimension objects...

...0142] A cube is a very precise definition of an OLAP cube that can be delivered using a single SQL statement. Each cube is derived from a single cube model. The cube facts and list of cube dimensions are subsets of those in the referenced cube model. A cube view name is also defined which represents the cube in the database. Cubes are appropriate for tools and applications that do not use multiple hierarchies because cube dimensions allow one cube hierarchy per cube dimension...

...0143] The purpose of a cube is to define a standard relational view of an OLAP structure. In addition to the relational view, a cube provides an extended describe (e.g., XML document) that describes the roles of its columns in multidimensional terms. In the process of

defining a cube , the designer selects a subset of the possible elements, choosing a single hierarchy for each dimension. This ensures that the cube unambiguously defines a single relational result set. The simplicity of a cube makes the cube useful to less sophisticated OLAP applications, such as portable devices powered by World Wide Web...

...0144] The metadata object specific properties of a cube metadata object are described in the following Table 12 in accordance with certain implementations of...

...0145] A cube facts metadata object has a subset of measures in an ordered list from a specific facts metadata object. A cube facts metadata object gives a cube the flexibility to scope facts for a cube model. The structural information, such as the joins and attributes, is referenced from the parent facts metadata object. The metadata object specific properties that define a cube facts metadata object are described in the following Table 13 in accordance with certain implementations...

...0146] A cube dimension metadata object is used to scope a dimension for use in a cube . The cube dimension metadata object references the dimension from which it is derived and the relevant cube hierarchy for the given cube . In certain implementations, one cube hierarchy can be applied to a cube dimension. The joins and attributes that apply to the cube dimension are referenced from the dimension definition. The metadata object specific properties that define a cube dimension metadata object are described in the following Table 14 in accordance with certain implementations...

...0147] A cube hierarchy metadata object is a scoped version of a hierarchy and is used in a cube . A cube hierarchy references the hierarchy from which it is derived and can have a subset of the attributes from the parent hierarchy. Additionally, a cube hierarchy metadata object references the attribute relationships that apply on the cube . In certain implementations, one cube hierarchy can be defined for a cube dimension of a cube . A cube hierarchy metadata object has the same hierarchy types and deployment mechanisms as the hierarchy from which the cube hierarchy metadata object is derived...

...0148] A cube hierarchy is very similar to a hierarchy; however, a cube dimension refers to a single cube hierarchy. This allows a single SELECT statement to calculate the cells of a cube .

[...

...0149] The metadata object specific properties that define a cube hierarchy metadata object are described in the following Table 15 in accordance with certain implementations...

...invention. The arrows indicate that a metadata object references another metadata object. For example, a cube metadata object 1610 references a cube model metadata object 1600. A more detailed relationship description of the metadata objects is illustrated...

...object rules for that metadata object. The rules are separated in three categories: Base Rules, Cube Model Completeness Rules, and Optimization Rules. The following discussion of specific rules provides a set...

...0154] The base rules for a cube model metadata object are: (1) the cube model metadata object refers to zero or one facts metadata object; (2) the cube model metadata object refers to zero or more dimension(s); (3) dimension-join pairs have...

...the dimension metadata object; and (5) for each measure referenced in the facts of the cube model, all the explicit dimension references in the aggregations of the measure are referenced by the cube model. When the cube model references at least one dimension, an aggregation with an empty dimension set matches to at least one dimension from the cube model that was not previously referenced...

...0155] The base rules for a cube metadata object are: (1) the cube metadata object refers to one cube facts; (2) the cube metadata object refers to at least one cube dimension; (3) cube facts is derived from the facts used in the cube model; and, (4) cube dimensions are derived from the dimensions used in the cube model...

...0158] The base rules for a cube facts metadata object are: (1) the cube facts metadata object refers to at least one facts; (2) the cube facts metadata object refers to at least one measure; and, (3) measures referenced by a cube facts metadata object are part of the facts metadata object...

...0159] The base rules for a cube dimension metadata object are as follows: (1) the cube dimension metadata object refers to one dimension; (2) the cube dimension metadata object refers to a cube hierarchy; and, (3) the cube hierarchy referenced by the cube dimension metadata object is derived from a hierarchy that is referenced by the dimension of the cube dimension metadata object...

...0161] The base rules for a cube hierarchy metadata object are: (1) the cube hierarchy metadata object refers to one hierarchy; (2) the cube hierarchy metadata object refers to at least one attribute; (3) attributes referenced by the cube hierarchy metadata object are part of the hierarchy; (4) the order of the attributes in the cube hierarchy metadata object are the same as in the hierarchy (with the exception of hierarchies...
...a left attribute as part of the hierarchy; and, (6) attribute relationships referenced in the cube hierarchy metadata object are also referenced in the hierarchy that defines the cube hierarchy...

...0166] The cube model completeness rules extend the base rules in order to ensure that a cube model has the required links to other metadata objects to allow effective warehouse SQL queries to be formed. The cube model completeness rules for a cube model metadata object are: (1) a cube model metadata object refers to one facts; (2) a cube model metadata object refers to one or more dimensions...

...0167] The optimization rules extend the cube model completeness rules in order to ensure that optimization of warehouse SQL queries can be...

...0168] The optimization rules for a cube model metadata object is: (1) the join used in the facts to dimension has a...

...properties in FIGS. 18A-18E are common properties. For example, FIGS. 18A-18E illustrate a cube model metadata object instance 1800, a cube metadata object instance 1802, a facts metadata object instance 1804, a cube facts metadata object instance 1806, measure metadata object instances 1808, 1810, dimension metadata object instances 1812, 1814, cube dimension metadata object instances 1816, 1818, hierarchy metadata object instances 1820, 1822, 1824, cube hierarchy metadata object instances 1826, 1828, join metadata object instances 1830, 1832, and attribute metadata...

...user may use the user interface 150 to create metadata objects. After creating an empty cube model metadata object, a facts metadata object

and dimension metadata objects are created and joined to the cube model metadata object by creating appropriate join metadata objects...the dimension set is empty, the aggregation function is applied to all dimensions in the cube or cube model that are not specifically being used by any other aggregations in the list...

...0180] The multidimensional metadata software 120 automatically generates a SQL statement for generation of a cube view using the metadata in the measure metadata object...

...a measure. The calculation order for the set of measure metadata objects referenced by a cube model metadata object or a cube metadata object need not be the same-each measure metadata object may specify a calculation...

...0185] Yet another requirement for measures is support for semi - additive measures , such as snapshot measures (e.g., Inventory). For example, for Market and Product dimensions, a...
...Product), (AVG, Time), (MAX, Market)). Targeting to sophisticated applications is a more generic representation of semi - additive measures , which are described further below with reference to FIG. 23. Also, the requirement of targeting...

...0192] The Cost measure metadata object 2210 is created for a cube model with the three dimensions: Product 2202, Market 2204, and Time 2206. The Cost measure...

...0193] The Revenue measure metadata object 2220 is created for a cube model with three dimensions: Product 2202, Market 2204, and Time 2206. The Revenue measure metadata...

...0194]FIG. 23 illustrates creation of a semi - additive measure in accordance with certain implementations of the invention. The Inventory measure metadata object 2310 has...

...Each hierarchy metadata object includes information used to build a ROLLUP clause. In particular, a cube model metadata object may reference several possible hierarchies, but selection of a single hierarchy is required for SQL generation, in certain implementations of the invention. In block 2904, a cube model description of a cube model metadata object that references the facts and one or more dimension metadata objects is received, and the cube model metadata object is generated from the cube model description. In block 2906, a selection of a subset of the cube model metadata object is received. In block 2908, a cube metadata object is generated based on the selection. Additionally, a SQL statement is generated for creating a cube view from metadata in one or more measure metadata objects. The generation of the SQL...

...in dimension metadata objects). A ROLLUP for each dimension returns results that represent an OLAP cube , in a relational way. The combination of more than one ROLLUP operator in a single...

...the following grouping clauses, which are a set of grouping clauses that make up a cube :

[...

...0227] Although FIG. 29A describes use of measure metadata objects to generate a cube view, in additional implementations, the measure metadata objects may be used without creating a cube view. That is, the cube view is one way to use the definitions of the measure metadata

objects in order...

...the measure metadata objects is for an application to read the measure definitions and the cube model metadata and to directly generate a SQL statement from the measure definitions and cube model metadata...

...logic for generating a SQL statement from one or more measure metadata objects and a cube model metadata object in accordance with certain implementations of the invention. Control begins in block...

...Each hierarchy metadata object includes information used to build a ROLLUP clause. In particular, a cube model metadata object may reference several possible hierarchies, but selection of a single hierarchy is required for SQL generation, in certain implementations of the invention. In block 2914, a cube model description of a cube model metadata object that references the facts and one or more dimension metadata objects is received, and the cube model metadata object is generated from the cube model description. In block 2916, a selection of a subset of the cube model metadata object is made by an application program. In block 2918, under control of the application program, using the cube model metadata object and one or more of the measure metadata objects, a SQL statement...

...120 handles various ,query types (e.g., Grand Total query, Slice based query, and complete cube query). In a Grand Total slice, only the grand total for all dimensions is returned. A slice is a sub- cube , while a complete cube is an entire cube .
[...

...have little or no symmetry) or specify conflicting calculation order for the dimensions of the cube , then those asymmetric measures are divided into subsets sharing the same calculation order and a...

...that retrieves multidimensional information. In certain implementations, the SQL statement may be catalogued as a cube view and referenced by a cube metadata object...in a report. However, the result of executing the SQL statement generated for a complete cube query is a cube view, which may itself be queried...

...0258] For a complete cube query with symmetric measures, the multidimensional metadata software 120 may generate the following select statement...

...in a report. However, the result of executing the SQL statement generated for a complete cube query is a cube view, which may itself be queried...

...0289] For a complete cube query with asymmetric measures, the multidimensional metadata software 120 may generate the following select statement...

...same set of attributes when these SQL statements are generated for the same slice or cube . Therefore, the attribute instances will be the same in all the SQL statements. The technique...

...depends on the type of SQL statements being combined (i.e., slice-based vs. complete cube). For slice-based SQL statements, the clause used in the inner join will use a...

...0344] For the complete cube type of SQL statement, the join clause also takes into consideration the fact that the...

...joins attribute instances when they contain specific members or contain

aggregation. The following is a cube -based SQL statement generated by the multidimensional metadata software 120 for the first option that...

...Machines Corporation in the United States and/or other countries. Windows is a trademark of Microsoft Corporation in the United States and/or other countries. Solaris and JDBC are trademarks of...

Exemplary or Independent Claim(s):

1. A method for specifying multidimensional calculations, comprising: receiving selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more...

...measure metadata objects; and generating a statement for retrieving multidimensional information using metadata in the cube model metadata object and the one or more measure metadata objects, wherein each of the...

...computer system having at least one program for: receiving selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more...

...measure metadata objects; and generating a statement for retrieving multidimensional information using metadata in the cube model metadata object and the one or more measure metadata objects, wherein each of the...

...causes operations to be performed, the operations comprising: receiving selection of a subset of a cube model metadata object that is generated from a facts metadata object and one or more...
...measure metadata objects; and generating a statement for retrieving multidimensional information using metadata in the cube model metadata object and the one or more measure metadata objects, wherein each of the...

Non-exemplary or Dependent Claim(s):

...statement further comprises: generating a SELECT statement for a slice of the subset of the cube model metadata object...

...structured query language statement further comprises: generating a SELECT statement for the subset of the cube model metadata object ...

...19. The method of claim 1, further comprising: generating a cube metadata object based on the selection of the subset of the cube model metadata object, including generating a structured query language statement for creation of a cube view, wherein the structured query language statement is generated from metadata in the one or...

...The method of claim 1, further comprising: under control of an application program, using the cube model metadata object and one or more of the measure metadata objects, generating a structured...

...at least one program further comprises: generating a SELECT statement for the subset of the cube model metadata object...

...at least one program further comprises: generating a SELECT statement for the subset of the cube model metadata object...

...The system of claim 21, wherein the at least one program further comprises: generating a cube metadata object based on the selection of the subset of the cube model metadata object, including

generating a structured query language statement for creation of a cube view, wherein the structured query language statement is generated from metadata in the one or...

...the at least one program further comprises: under control of an application program, using the cube model metadata object and one or more of the measure metadata objects, generating a structured...

...statement further comprise: generating a SELECT statement for a slice of the subset of the cube model metadata object...

...structured query language statement further comprise: generating a SELECT statement for the subset of the cube model metadata object ...

...59. The article of manufacture of claim 41, the operations further comprising: generating a cube metadata object based on the selection of the subset of the cube model metadata object, including generating a structured query language statement for creation of a cube view, wherein the structured query language statement is generated from metadata in the one or...

...of claim 41, the operations further comprising: under control of an application program, using the cube model metadata object and one or more of the measure metadata objects, generating a structured...

Dialog eLink: Order File History
2/3,K/5 (Item 2 from file: 654)
DIALOG(R)File 654: US PAT.FULL.

0005544179 **IMAGE Available
Derwent Accession: 2004-213729
Universal drill-down system for coordinated presentation of items in different databases
Inventor: Thomson, Neil, INV
Paiement, Andre, INV
Gould, Dave, INV
Peticlerc, Martin, INV
Donnelly, Brian, INV
Chow, Gordon, INV
Assignee: Business Objects S.A. 03), Levallois-Perret, FR, 157/159 rue Anatole France 92309
Correspondence Address: TOWNSEND AND TOWNSEND AND CREW, LLP, TWO EMBARCADERO CENTER EIGHTH FLOOR, SAN FRANCISCO, CA, 94111-3834, US

	Publication Number	Kind	Date	Application Number	Filing Date
Main Patent	US 20040034615	A1	20040219	US 2002321781	20021216
Provisional				US 60-341651	20011217

Fulltext Word Count: 18448

**IMAGE Available

Summary of the Invention:

...languages, are provided by, e.g., different manufacturers such as Oracle, International Business Machines (IBM), Microsoft , etc. One characteristic that relational databases provide (even among different types) is the ability to...

...partitions" as a mechanism to allow OLAP queries to access the relational data under a cube . Unfortunately, regardless of the technology employed these prior art systems are limited in the number...

Description of the Invention:

...database. In most cases, a database has a hierarchy of frameworks (e.g. server/ database/ cube / dimension/ level) as data can be complex and large...

...0047] "Value" denotes any additive or semi - additive numerical or measure value in a report. In case of a cell, a value cell contains a numerical...The WebI OLAP implementation uses a browser 402 or Internet Explorer, which is available from Microsoft Corporation. Browser 402 is used to display WebI OLAP documents in the client environment 118...

...0086] In the server environment, Microsoft ’s Web hosting platform marketed as the Internet Information Services (IIS 4/5) 404 is...

...maps 504 and includes the capability to use selected application program interfaces (APIs) to access cube and relational metadata and to provide the originating and target data source metadata. The designer 502 acquires OLAP metadata from metadata model 508. Model 508 acquires data from OLAP cube 510 through requests handled by access engine 512 in a manner understood in the art...

...snap shot that shows an actual working translation model open for modification with the OLAP (cube) originating data source in the top left pane and the relational (Universe) data source in...

...preferred embodiment, the UDS Designer is written in Visual Basic (VB), which is available from Microsoft Corporation, on top of a UDS SDK. As illustrated in FIG. 7, the UDS Designer...given originating structure. In general, when translating from an OLAP to Relational source, the OLAP cube Time dimension has a Year Level and the Relational source has a Universe Class of...

...for any members. For example, the administrator may prohibit translation of any members from an Accounts dimension. In another instance, the administrators may prohibit users drill-through for the top two levels in a cube and force drill-through to occur at lower level members only. The disable structure translation...

...0145] Where OLAP cube members are the same as the Universe Object’s Values (France=France), the Universe Object...

...be qualified. This technique is referred to as structure mapping. For mapping from an OLAP cube to another data source, this is also referred to as level mapping. This scenario is usually only possible where the OLAP cube as constructed from a well-formed Relational Star Schema or a specific drill-through Relational...

...UDS Designer because it only needs a single translation for all members for given OLAP Cube Level...

...Values that correspond to the OLAP Members are not unique and is frequently encountered in Microsoft Analysis Services cubes or when drilling from a relational source. Accordingly, there is the need to

translate additional...

...0150] OLAP cubes often have members that do not translate either directly or indirectly to a relational data...

...needed for each member that needs to be "disabled". A specific example for an Essbase cube is to disable passing a Dynamic Time Series member-YTD (Year to Date) is illustrated...

...0154] When a calculated member in an OLAP cube does not map directly to a column in a relational database or when drilling through...

...0158] Employees or part numbers or account dimensions are frequently based on relational parent/child table structures. While an OLAP Cube user may see multiple levels, there is internally only one level at the relational source...

...qualify a value for a column in terms of how it maps to an OLAP cube . This either requires that the Relational drill-through extraction forces the user to create a...

...the user will have to be prompted to select the specific member in the OLAP cube .

...

...0246] // error-should have found a dimension-map must be out of sync with actual cube Throw Error(Dimension not found...0262] // error-should have found a Hierarchy-map must be out of sync with actual cube Throw Error(Hierarchy not found...

...0279] // error-should have found a Level-map must be out of sync with actual cube Throw Error(Level not found...

...list of target reports, from target list reports 1210, that are valid for the OLAP cube that report 1222 is based on. Client 1218 presents the list of the reports to...

Exemplary or Independent Claim(s):

Non-exemplary or Dependent Claim(s):
...The database system of claim 26 further comprising means for defining translation maps for accessing cube metadata if said first or second data source is an OLAP data source and relational...

? t s2/9,k/1

Dialog eLink: USPTO Full Text Retrieval Options

2/9,K/1 (Item 1 from file: 15)
DIALOG(R)File 15: ABI/Inform(R)

02743296 477856671
Applying UML and XML for designing and interchanging information for data warehouses and OLAP applications

Trujillo, Juan; Lujan-Mora, Sergio; Song, Il-Yeol
Journal of Database Management v15n1 pp: 41-72
Jan-Mar 2004
CODEN: JDAMEQ
ISSN: 1063-8016 **Journal Code:** DAN
Document Type: Periodical; Feature **Language:** English **Record Type:** Fulltext
Special Feature: Chart Table Illustration Formula
Word Count: 10371

Abstract:

Multidimensional (MD) modeling is the basis for data warehouses, multidimensional databases (MDB) and on-line analytical processing (OLAP) applications. This paper presents how the unified modeling language (UML) can be successfully used to represent both structural and dynamic properties of these systems at the conceptual level. The structure of the system is specified by means of a UML class diagram that considers the main properties of MD modeling with minimal use of constraints and extensions of the UML. If the system to be modeled is too complex, the paper describes how to use the package grouping mechanism provided by the UML to simplify the final model. Furthermore, a UML-compliant class notation (called cube class) is provided to represent OLAP users' initial requirements. The paper also describes how to use the UML state and interaction diagrams to model the behavior of a data warehouse system. To facilitate the interchange of conceptual MD models, a Document Type Definition (DTD) is provided to represent the same MD modeling properties that can be considered by using our approach. From this DTD, valid extensible Markup Language (XML) documents can be directly generated that represent MD models at the conceptual level.

Text:

ABSTRACT

Multidimensional (MD) modeling is the basis for data warehouses (DW), multidimensional databases (MDB) and on-line analytical processing (OLAP) applications. In this paper, we present how the unified modeling language (UML) can be successfully used to represent both structural and dynamic properties of these systems at the conceptual level. The structure of the system is specified by means of a UML class diagram that considers the main properties of MD modeling with minimal use of constraints and extensions of the UML. If the system to be modeled is too complex, thereby leading us to a considerable number of classes and relationships, we describe how to use the package grouping mechanism provided by the UML to simplify the final model. Furthermore, we provide a UML-compliant class notation (called cube class) to represent OLAP users' initial requirements. We also describe how we can use the UML state and interaction diagrams to model the behavior of a data warehouse system. To facilitate the interchange of conceptual MD models, we provide a Document Type Definition (DTD) which allows us to represent the same MD modeling properties that can be considered by using our approach. From this DTD, we can directly generate valid extensible Markup Language (XML) documents that represent MD models at the conceptual level. We believe that our innovative approach provides a theoretical foundation for simplifying the conceptual design of MD systems and the examples included in this paper clearly illustrate the use of our approach.

Keywords: data warehouses; multidimensional databases; OLAP; conceptual modeling; UML; object orientation; ODBMS; XML

INTRODUCTION

Multidimensional (MD) modeling is the foundation for data warehouses (DW), multidimensional databases (MDB) and on-line analytical processing (OLAP) applications. The benefit of using MD modeling is two-fold. On one hand, the MD model is close to data analyzers' way of thinking; therefore, it helps users understand data. On the other hand, the MD model supports performance improvement as its simple structure allows us to predict final users' intentions.

Some approaches have been proposed lately to accomplish the conceptual design of these systems. Unfortunately, none of them have been accepted as a standard for DW conceptual modeling. These proposals try to represent main MD properties at the conceptual level with special emphasis on MD data structures. A conceptual modeling approach for DW, however, should also concern other relevant aspects such as users' initial requirements, the behavior of the system (e.g., main operations to be accomplished on MD data structures), available data sources, specific issues for automatic generation of the database schema and so on. We claim that object orientation with the UML provides an adequate notation for modeling every aspect of a DW system (MD data structures, the behavior of the system, etc.) from user requirements to implementation.

We have proposed an object-oriented (OO) approach to accomplish the conceptual modeling of DW, MDB and OLAP applications that introduces a set of minimal constraints and extensions of the UML (Unified Modeling Language) (Booch, Rumbaugh, and Jacobson, 1998; OMG, 2001), needed for an adequate representation of MD modeling properties (Trujillo, 2001; Trujillo, Palomar, Gomez, and Song, 2001b). These extensions are based on the standard mechanisms provided by the UML to adapt it to a specific method or model (e.g., constraints, tagged values). We have also presented how to group classes into packages to simplify the final model in case the model becomes too complex due to the high number of classes (Lujan-Mora, Trujillo, and Song, 2002). Furthermore, we have provided a UML-compliant class notation to represent OLAP users' initial requirements (called cube class). From these cube classes, we then describe the use of state and interaction diagrams to model the behavior of the system based on the applied OLAP operations (Trujillo, Palomar, and Gomez, 2000). We have also discussed issues such as identifying attributes and descriptor attributes that set the basis for an adequate semi-automatic generation of a database schema and user requirements in a target commercial OLAP tool.

The UML can also be used with powerful mechanisms such as the Object Constraint Language (OCL) (Warmer and Kleppe, 1998; OMG, 2001) and the Object Query Language (OQL) (Cattell et al., 2000) to embed DW constraints (e.g. additivity and derived attributes) and users' initial requirements in the conceptual model. In this way, when we model a DW system, we can obtain simple yet powerful extended UML class diagrams that represent main MD properties at a conceptual level.

On the other hand, a salient issue these days in the scientific community and in the business world is the interchange of information. The extensible Markup Language (XML) (W3C, 2000) is rapidly being adopted as the standard syntax for the interchange of un-structured, semi-structured and structured data. XML is an open neutral platform and vendor independent meta-language, which allows us to reduce the cost, complexity, and effort required in integrating data within and between enterprises.

From these considerations, in this paper we present the following contributions. We believe that our innovative approach provides a theoretical foundation for the possible use of Object-Oriented Databases (OODB) and Object-Relational Databases (ORDB) for DW and OLAP applications. For this reason, we provide the representation of our approach into the standard for OODB proposed by the Object Database Management Group (ODMG) (Catell et al. 2000). We also believe that a relevant feature of a conceptual model should be its capability to share information in an easy and standard form. Therefore, we also present how to represent MD models, accomplished by using our approach based on the UML, by means of the XML. In order to do this, we provide a Document Type Definition (DTD) that defines the correct structure and content of an XML document representing main MD properties. Finally, to show the benefit of our approach, we include a set of case studies to show the elegant way in which our proposal represents both structural and dynamic properties of MD modeling.

The remainder of this paper is organized as follows. The next section details the major features of MD modeling that should be taken into account for a proper MD conceptual design. Then the paper summarizes the most relevant conceptual approaches proposed so far by the research community, and we present how we use the UML to consider main structural and dynamic MD properties at the conceptual level. We also present how to facilitate the interchange of MD models by generating the corresponding standard provided by the ODMG and the DTD from UML. We present a set of case studies taken from Kimball (Kimball and Ross, 2002) to show the benefit of our approach and finally, the last section draws conclusions and sketches out new research that is currently being investigated.

MULTIDIMENSIONAL MODELING

In MD modeling, information is structured into facts and dimensions. A fact is an item of interest for an enterprise, and is described through a set of attributes called measures or fact attributes (atomic or derived), which are contained in cells or points in the data cube. This set of measures is based on a set of dimensions that determine the granularity adopted for representing facts (i.e., the context in which facts are to be analyzed). Moreover, dimensions are also characterized by attributes, which are usually called dimension attributes. They are used for grouping, browsing and constraining measures.

Let us consider an example in which the fact is the product sales in a large store chain and the dimensions are as follows: product, store, customer and time. On the left hand side of Figure 1, we can observe a data cube typically used for representing an MD model. In this particular case, we have defined a cube for analyzing measures along the product, store and time dimensions.

We note that a fact usually represents a many-to-many relationship between any of two dimensions. For example, a product is sold in many stores and a store sells many products. We also assume that there is a many-to-one relationship between a fact and each particular dimension. For example, for each Store there are many sale tickets, but each sale ticket belongs to only one store.

Nevertheless, there are some cases in which a fact may be associated with a particular dimension as a many-to-many relationship. For example, the fact product_sales is considered as a particular many-to-many relationship to the product dimension, as one ticket may consist of more than one product even though every ticket is still purchased in only one store by one customer and at one time.

With reference to measures, the concept of additivity or summaribility

(Blaschka, Sapia, Hofling, and Dinter, 1998; Golfarelli, Maio, and Rizzi, 1998; Kimball and Ross, 2002; Trujillo et al., 2001b), (Tryfona, Busborg, and Christiansen, 1999) on measures along dimensions is crucial for MD data modeling. A measure is additive along a dimension if the SUM operator can be used to aggregate attribute values along all hierarchies defined on that dimension. The aggregation of some fact attributes (roll-up, in OLAP terminology), however, might not be semantically meaningful for all measures along all dimensions. A measure is semi-additive if the SUM operator can be applied to some dimensions, but not all the dimensions. A measure is non-additive if the SUM operator cannot be applied to any dimension. In our example, number of clients (estimated by counting the number of purchased receipts for a given product, day and store) is not additive along the product dimension. Since the same ticket may include other products, adding up the number of clients along two or more products would lead to inconsistent results. However, other aggregation operators (e.g. SUM, AVG and MIN) could still be used along other dimensions such as time. Thus, number of clients is semi-additive. Finally, examples of non-additive measures would be those measures that record a static level such as inventory, financial account balances or measures of intensity such as room temperatures (Kimball and Ross, 2002).

Figure 1: A data cube and classification hierarchies defined on dimensions

Regarding dimensions, the classification hierarchies defined on certain dimension attributes are crucial because the subsequent data analysis will be addressed by these classification hierarchies. A dimension attribute may also be aggregated (related) to more than one hierarchy. Therefore, multiple classification hierarchies and alternative path hierarchies are also relevant. For this reason, a common way of representing and considering dimensions with their classification hierarchies is by means of Directed Acyclic Graphs (DAG).

On the right hand side of Figure 1, we can observe different classification hierarchies defined on the product, Store and time dimensions. On the product dimension, we have considered a multiple classification hierarchy to be able to aggregate data values along two different hierarchy paths: (i) product, type, family, group and (ii) product, brand. On the other hand, we can also find attributes that are not used for aggregating purposes, instead they provide features for other dimension attributes (e.g. product_name). On the Store dimension, we have defined an alternative classification hierarchy with two different paths that converge into the same hierarchy level: (i) store, city, province, state and (ii) store, sales_area, state. Finally, we have defined another multiple classification hierarchy with the following paths on the time dimension: (i) time, month, semester, year and (ii) time, season.

Nevertheless, classification hierarchies are not so simple in most cases. The concepts of strictness and completeness are quite important, not only for conceptual purposes, but also for further design steps of MD modeling (Tryfona et al.). Strictness means that an object of a lower level in a hierarchy belongs to only one in a higher level, e.g. a province is only related to one state. Completeness means that all members belong to one higher-class object which consists of those members only. For example, suppose that the classification hierarchy between the state and province levels is "complete". In this case, a state is formed by all the provinces recorded and all the provinces that form the state are recorded.

OLAP scenarios sometimes become very large as the number of dimensions increases significantly, which may then lead to extremely sparse dimensions and data cubes. In this way, there are some attributes that are normally valid for all elements within a dimension while others are only valid for a subset of elements (also known as the categorization of dimensions (Lehner,

1998; Tryfona et al.,n 1999). For example, attributes alcohol percentage and volume would only be valid fordrink_products and will be "null" for food_products. Thus, a proper MD data model should be able to consider attributes only when necessary, depending on the categorization of dimensions.

Furthermore, let us suppose that apart from a high number of dimensions (e.g. 20) with their corresponding hierarchies, we have a considerable number of facts (e.g. 8) sharing dimensions and classification hierarchies. This system will lead us to a very complex design, thereby increasing the difficulty in reading the modeled system. To avert a convoluted design, an MD conceptual model should also provide techniques to avoid flat diagrams, allowing us to group dimensions and facts to simplify the final model.

Once the structure of the MD model has been defined, OLAP users usually define a set of initial requirements as a starting point for the subsequent data analysis phase. From these initial requirements, users can apply a set of operations (usually called OLAP operations) (Chaudhuri and Dayal, 1997) to the MD view of data for further data analysis. These OLAP operations are usually as follows: roll-up (increasing the level of aggregation) and drill-down (decreasing the level of aggregation) along one or more classification hierarchies, slice-dice (selection and projection) and pivoting (re-orienting the MD view of data which also allows us to exchange dimensions for facts; i.e. symmetric treatment of facts and dimensions).

Star Schema

In this sub-section, we will summarize the star schema popularized by Kimball (Kimball and Ross, 2002), as it is the most well-known schema representing MD properties in relational databases.

Kimball claims that the star schema and its variants fact constellations schema and the snowflake schema are logical choices for MD modeling to be implemented in relational systems. We will briefly introduce this well-known approach using Sales Dimensional Model.

Figure 2 shows an example of Kimball’s Sales Dimensional Model. In this model, the fact is the name of the middle box (Sales fact table). Measures are the non-foreign keys in the fact table (dollars_sold, units_sold, and dollars_cost). Dimensions are the boxes connected to the fact table in a one-to-many relationship (Time, Store, Product, Customer, and Promotion). Each dimension contains relevant attributes: day_of_week, week_number, and month in Time; store_name, address, district, and floor_type in Store, and so on.
From Figure 2, we can easily see that there are many MD features that are not reflected in the Dimensional Model: Which are the classification hierarchies defined on dimensions? Can we use all aggregation operators on all measures along all dimensions? What are these classification hierarchies like (non-strict, strict, and complete)? And many more properties. Therefore, we argue that for a proper DW and OLAP design, a conceptual MD model should be provided to better reflect user requirements. This conceptual model could then be translated into a logical model for a later implementation. In this way, we can be sure that we are analyzing the real world as users perceive it.

Figure 2: Sales Dimensional Model

RELATED WORK

Lately, several MD data models have been published. Some of them fall into the logical level (such as the well-known star schema by Kimball (Kimball and Ross, 2002). Others may be considered as formal models, as they provide

a formalism to consider main MD properties. A review of the most relevant logical and formal models can be found in Blaschka et al. and Abello, Samos and Saltor (2001).

In this section, we will only briefly make reference to the most relevant models that we consider "pure" conceptual MD models. These models provide a high level of abstraction for the main MD modeling properties presented previously and are totally independent from implementation issues. These are as follows: The Dimensional-Fact (DF) model by Golfarelli et al. (1998), The Multidimensional/ER (M/ER) model by Sapia, Blaschka, Hofling, and Dinter (1998) and Sapia (1999) and The starER model by Tryfona et al. (1999).

In Table 1, we provide the coverage degree of each above-mentioned conceptual model regarding the main MD properties described in the previous section. To start with, to the best of our knowledge, no proposal provides a grouping mechanism to avoid flat diagrams and to simplify the conceptual design when a system becomes complex due to a high number of dimensions and facts sharing dimensions and their corresponding hierarchies. Regarding facts, only the starER model considers many-to-many relationships between facts and particular dimensions by indicating the exact cardinality (multiplicity) between them. None of them consider derived measures or their derivation rules as part of the conceptual schema. The DF and the starER models consider the additivity of measures by explicitly representing the set of aggregation operators that can be applied on non-additive measures. With reference to dimensions, all of the models consider multiple and alternative path classification hierarchies by means of Directed Acyclic Graphs defined on certain dimension attributes. However, only the starER model considers non-strict and complete classification hierarchies by specifying the exact cardinality between classification hierarchy levels. As both the M/ER and the starER models are extensions of the Entity Relationship (ER) model, they can easily consider the categorization of dimensions by means of Is-a relationships.

Table 1: Comparison of conceptual multidimensional models

With reference to the dynamic level of MD modeling, the starER model is the only one that does not provide an explicit mechanism to represent users' initial requirements. On the other hand, only the M/ER model provides a set of basic OLAP operations to be applied from these users' initial requirements, and it models the behavior of the system by means of state diagrams.

We note that all the models provide a graphical notation that facilitates the conceptual modeling task to the designer. On the other hand, only the M/ER model provides a framework for an automatic generation of the database schema into a target commercial OLAP tool (particularly into Informix Metacube and Cognos Powerplay).

Finally, none of the proposals from Table 1 provide a mechanism to facilitate the interchange of the models following standard representations. Regarding MD modeling and the extensible Markup Language (XML) (W3C, 2000), some proposals have been presented. All of these proposals make use of XML as the base language for describing data. In Pokorny (2001), an innovative data structure called an XML-star schema is presented with explicit dimension hierarchies using DTDs that describe the structure of the objects permitted in XML data. The approach presented in Golfarelli, Rizzi and Vrdoljak (2001) proposes a semi-automatic approach for building the conceptual schema for a data mart starting from the XML sources. However, these approaches focus on the presentation of the multidimensional XML data rather than on the presentation of the structure of the multidimensional conceptual model itself.

From Table 1, one may conclude that none of the current conceptual modeling approaches consider all MD properties at both the structural and dynamic levels. Therefore, we claim that a standard conceptual model is needed to consider all MD modeling properties at both the structural and dynamic levels. We argue that an OO approach with the UML is the right way of linking structural and dynamic level properties in an elegant way at the conceptual level.

MULTIDIMENSIONAL MODELING WITH UML

In this section, we summarize how our OO MD model, based on a subset of the UML, can represent main structural and dynamic properties of MD modeling. First we will specify main structural properties by means of a UML class diagram. Secondly, we summarize how users' initial requirements are easily considered by what we call cube classes. We next describe how to model the behavior of MD databases by using UML state and interaction diagrams from the information represented in these cube classes. The next section sketches how we automatically transform an MD model accomplished by following our approach into the Object Database Standard defined by the Object Database Management Group (ODMG) (Cattell et al., 2000). Finally, we present the corresponding representation of our approach into the XML (W3C, 2000) to allow an easy interchange of MD information.

Structural Properties by Using UML Class Diagrams

The main structural features considered by UML class diagrams are the many-to-many relationships between facts and dimensions, degenerate dimensions, multiple and alternative path classification hierarchies, and non-strict and complete hierarchies.

It is important to remark that if we are modeling complex and large DW systems, we are not restricted to using flat UML class diagrams. Instead, we can make use of the grouping mechanism provided by the UML called package to group classes together into higher level units to create different levels of abstraction, therefore simplifying the final model (Lujan et al., 2002). In this way, a UML class diagram improves and simplifies the system specification accomplished by classic semantic data models such as the ER model. Furthermore, necessary operations and constraints (e.g. additivity rules) can be embedded in the class diagram by means of OCL expressions (Warmer and Kleppe, 1998; OMG, 2001).

In this approach, the main structural properties of MD models are specified by means of a UML class diagram in which the information is clearly separated into facts and dimensions. Dimensions and facts are represented by dimension classes and fact classes, respectively. Then, fact classes are specified as composite classes in shared aggregation relationships of n dimension classes. The flexibility of shared aggregations in the UML allows us to represent many-to-many relationships between facts and particular dimensions by indicating the 1..* cardinality on the dimension class role. In our example in Figure 3 (a), we can see how the fact class Sales has a many-to-one relationship with both dimension classes.

Figure 3: Multidimensional modeling using UML

By default, all measures in the fact class are considered additive. For non-additive measures, additivity rules are defined as constraints and are included in the fact class. Furthermore, derived measures can also be explicitly considered (indicated by /) and their derivation rides are placed between braces near the fact class, as shown in Figure 3(a).

This OO approach also allows us to define identifying attributes in the

fact class, by placing the constraint {OID} next to an attribute name. In this way we can represent degenerate dimensions (Giovinazzo, 2000; Kimball and Ross, 2002), thereby representing other fact features in addition to the measures for analysis. For example, we could store the ticket number (ticket_num) and the line number (line_num) as degenerate dimensions, as reflected in Figure 3(a).

With respect to dimensions, every classification hierarchy level is specified by a class (called a base class). An association of classes specifies the relationships between two levels of a classification hierarchy. The only prerequisite is that these classes must define a Directed Acyclic Graph rooted in the dimension class (constraint {dag} placed next to every dimension class). The DAG structure can represent both alternative path and multiple classification hierarchies. Every classification hierarchy level must have an identifying attribute (constraint {OID}) and a descriptor attributel (constraint {D}). These attributes are necessary for an automatic generation process into commercial OLAP tools, as these tools store the information in their metadata. The multiplicity 7 and 1..*, defined in the target associated class role, addresses the concepts of strictness and non-strictness, respectively. Strictness means that an object at a hierarchy's lower level belongs to only one higher-level object (e.g., as one month can be related to more than one season, the relationship between them is non-strict). Moreover, defining the {completeness} constraint in the target associated class role addresses the completeness of a classification hierarchy (see an example in Figure 3(b)). By completeness we mean that all members belong to one higher-class object and that object consists of those members only. For example, all the recorded seasons form a year, and all the seasons that form the year have been recorded. Our approach assumes all classification hierarchies are non-complete by default.

Finally, the categorization of dimensions, used to model additional features for a class's subtypes, is represented by means of generalization-specialization relationships. However, only the dimension class can belong to both a classification and a specialization hierarchy at the same time. An example of categorization for the Product dimension is shown in Figure 3(c).

Dynamic Properties

Regarding dynamic properties, this approach allows us to specify users' initial requirements by means of a UML-compliant class notation called cube class. After requirements are specified, behavioral properties are usually then related to these cube classes that represent users' initial requirements. We particularly use state and interaction diagrams to model the behavior (evolution) of these cube classes based on the applied OLAP operation.

Cube classes follow the query-by-example (QBE) method: the requirements are defined by means of a template with blank fields. Once requirements are defined, the user can then enter conditions for each field that are included in the query. We provide a graphical representation to specify users' initial requirements because QBE systems are considered easier to learn than formal query languages. The structure of a cube class is shown in Figure 4:

* Cube class name.

* Measures area, which contains the measures from the fact to be analyzed.

* Slice area, which contains the constraints to be satisfied in the dimensions.

* Dice area, which contains the dimensions and their grouping conditions to address the analysis.

* Order area, which specifies the order of the result set.

* Cube operations, which cover the OLAP operations for a further data-analysis phase.

We should point out that this graphical notation of the cube class aims at facilitating the definition of users' initial requirements to non-expert UML or databases users. In a more formal way, every one of these cube classes has its underlying OQL specification. Moreover, an expert user can directly define cube classes by specifying the OQL sentences (see more details on the representation of cube classes further in the paper).

Figure 4: Cube class structure

Behavioral Properties by Using State and Interaction Diagrams

From these cube classes seen in the previous section, final users may start a navigational process by applying certain OLAP operations (roll-up, drill-down, etc.) in the further data analysis phase. These operations are closed as they generate another cube class as an output. Thus, we use state and interaction diagrams2 to model the behavior (evolution) of these cube classes based on the applied OLAP operation. These diagrams contain information about the most probable evolution of final users' requirements from the specified initial requirement. The information contained in these diagrams can be used by OLAP designers to predict user behaviors, and therefore, help them design a proper view maintenance policy.

Regarding state diagrams, one state diagram is defined for each initial cube class. The diagram specifies that certain OLAP operations lead users to cube classes that allow them to analyze the same data (the same measures along the same dimensions) in different ways (navigating through the classification hierarchies defined along the dimensions considered). In these diagrams, each classification hierarchy level defined on a dimension included in the Dice area is considered as a valid state. Every one of these valid states will be a new cube class. Then, the provided OLAP operations allow us to navigate along the states to define new cube classes.

In Figure 5, we can see an example of state diagrams. One state is defined for every level considered in the classification hierarchy of the dimensions included in the corresponding Dice area of the cube class. The data analysis will start on the initial state that corresponds to the finest condition specified in the Slice area. Let us suppose that we are interested in navigating along both the product and store dimensions. Classical roll-up and drill-down OLAP operations will allow us to aggregate and de-aggregate data (measures) respectively along the hierarchy levels defined in the classification hierarchies. Finally, from every state, we can finish the data analysis with the destroy operation that will lead us to the final state.

On the other hand, an interaction diagram can also be defined for each UML class diagram. In our approach, we have adopted sequence diagrams (Booch et al. 1998; OMG, 2001) for their clarity and low complexity. This interaction diagram shows interactions among cube classes, changed by OLAP operations such as rotate, pivot, slice, or dice. Thus, we can specify that certain OLAP operations (e.g., dice) lead users to cube classes which will show completely different data. Thus, these new cube classes represent the most probable new requirements a final user wishes to execute.

In Figure 6, we can see an example of interaction diagrams. Let us suppose that we have only defined two cube classes to specify two initial requirements. Then, we specify the operation needed to switch from one cube class into the other. In this particular case, the rotate operation indicates the transition that lead us to the CC_2 from the CC_1. In concrete, we are also interested in analyzing data along the Customer dimension. As we are still interested in the dimensions defined in CC_1, we do not eliminate any dimension in this operation. It is easy to see that we always define the reverse operation to give analyzers the opportunity of returning to the initial point of analysis.

Figure 5: An example of state interaction diagram

Standard Representation by Using the ODMG Proposal

Figure 6: An example of interaction diagrams

Our approach generates the corresponding representation of an MD model in most of the relational database management systems such as Oracle, Informix, Microsoft SQL Server, IBM DB2 and so on (Trujillo et al., 2001b). Furthermore, we also provide the corresponding representation into object-oriented databases. However, this representation is totally dependent on the object database management system (ODBMS) used for the corresponding implementation. For this reason, in this section we present the corresponding representation of an MD model accomplished by our approach following the standard for ODBMS3 prpposed by the Object Database Management Group (ODMG) (Cattell et al., 2000). The adoption of this standard ensures the portability of our MD model across platforms and products, thereby facilitating the use of our approach. However, we also point out some properties that cannot be directly represented by using this standard and that should be taken into account when transforming this ODBM into a particular object-oriented model of the target ODBMS.

The major components of the ODMG standard are the Object Model, the Object Definition Language (ODL), the Object Query Language (OQL), and the bindings of the ODMG implementations to different programming languages (C++, Smalltalk and Java). In this paper, we will start by providing the corresponding representation for structural properties into the ODL, a specification language used to define the specifications of object types. Then, we will sketch how to represent cube classes into the OQL, a query language that supports the ODMG data model. The great benefit of this OQL is that it is very close to SQL, and is therefore, a very simple-to-use query language.

ODL Definition of an MD Model
Our three-level MD model cannot be represented in an ODBMS, because the ODL uses a flat representation for the class diagram without providing any package mechanism in the ODL. Therefore, we start the transformation of the MD models from the third level in the fact package, because it contains the complete MD model definition: fact classes, dimension classes, base classes, classification hierarchy properties, etc.

In the following, we are going to use an actual example to clarify our approach. We have selected a simplification of the grocery example taken from Kimball's book (Kimball and Ross, 2002). In this example, the corresponding MD model contains the following elements:

Figure 7: Level 1 of the Grocery example

* One fact (Sales) with three measures (quantity, price and total_price) and two degenerate dimensions (ticket_num and line_num).

* Two dimensions: Product, with three hierarchy levels (brand, subgroup, and group) and time, with two hierarchy levels (month and year).

The first level of the MD model is represented in Figure 7 and only contains one star schema package, as the example only contains one fact. The second level contains one fact package (Sales product) and two dimension packages (Product and Time), as it can be seen in Figure 8. Finally, Figure 9 represents the content of the Product dimension package, and Figure 10 the content of the Time dimension package.

In Figure 11, we can see the content (level 3) of the Sales products fact package, where the complete definition of the MD model is available. The transformation process starts from this view of the MD model.

For the sake of simplicity, we show the ODL representation for only three classes: Sales, Product, and Time (the representation of the other classes is very similar). The transformation process starts from the fact class (Sales). Since OID attributes cannot be represented in ODL, we have decided to use the unsigned long type to represent them. Aggregation relationships cannot be directly represented, but we transform them to association relationships. Moreover, maximum cardinality of relationships can be expressed, but the minimum cardinality is lost in the transformation process. In ODL, the definition of a relationship includes designation of the target type, the cardinality on the target side, and information about the inverse relationship found in the target side. The ODL definition for the Sales fact class is as follows:

Figure 9: Level 3 for the Product dimension of the Grocery example

Figure 10: Level 3 for the Time dimension of the grocery example
Figure 11: Level 3 of the Sales fact of the Grocery example

For expressing the cardinality ?-to-many, we use the ODL constructor set. For example, the Product class has three relationships: with Sales class (?-to-many), with Brand class (?-to-one) and with Subgroup class (?-to-many). In order to know the cardinality of the relationships in this side, we have to consult the inverse relationship in the target side. For example, the relationship between Product and Sales is one-to-many, since the type of relationship is set (many) on this side, but in the inverse relationship (Sales::sales_product) it is Product (one). Product and Time dimension classes are specified in ODL as:

Loss of Expressiveness

As previously noted, some MD properties that are captured in our approach cannot be directly considered by using ODL. This is an obvious problem, because the ODL is a general definition language that is not oriented to represent MD properties used in a conceptual design. Specifically, we ignore or transform the following properties:

* Identifying attribute (OID) and descriptor attribute (D) are ignored because they are considered to be an implementation issue that will be automatically generated by the ODBMS.
* Initial values are ignored. This is not a key issue in conceptual MD modeling.

* Derived attributes and their corresponding derivation rules are ignored. These derivation rules will have to be specified when defining user requirements by using the OQL.

* Additivity rules are ignored because the ODL specification cannot

represent any information related to the aggregation operators that can be applied on measures.

* Minimum cardinality cannot be specified either.

* Completeness of a classification hierarchy is also ignored.

Up to now, these ignored properties have to be considered as footnotes in the ODMG specification. For an unambiguous specification of MD models using the ODMG specification, a formal constraint language should be used. Unfortunately, a constraint language is completely missing from the ODMG standard specification.

Cube Classes Represented by Using OQL

The OQL is not easy to use for defining users' initial requirements, because the user needs to know the underlying ODL representation corresponding to the MD model. Due to this fact, we also provide cube classes, which allow the user to define initial requirements in a graphical way. These cube classes can automatically be transformed into OQL sentences, and can therefore be used to query an ODBMS that stores an MD model. For example, let us suppose the following initial requirement:

The quantity sold of the products belonging to the "Grocery" Group during "January", grouped according to the product Subgroup and the Year and ordered by the Brand of the product

In Figure 12, we can see the corresponding cube class to the previous requirement. It is easy to see how the cube class is formed:

* Measures contains the goal of the analysis: SUM(quantity).

* Slice the restrictions defined on the Time and Product dimensions.

* Dice the grouping conditions required along the Product and Time dimensions.

* And Order defines the order of the result set.

The cube class can be automatically translated into OQL. The algorithm uses the corresponding ODL definition of the MD model to obtain the paths from the fact class (the core of the analysis) to the rest of classes (dimension and base classes). For example, the path from the Sales fact class to the Year base class along the Time dimension traverses the relationships sales_time in Sales fact class, time_month in Time dimension class, and month_year in Month base class. Moreover, when attributes' names are omitted in the cube class, the algorithm automatically selects the descriptor attribute defined in the MD model. For example, the expression Time.Month= "January" of the cube class in Figure 12 involves the use of the descriptor attribute from the Month base class, because no further attribute is specified. In the same way, the order expression Product.Brand involves the use of the descriptor attribute from Brand. The OQL for the corresponding cube class in Figure 12 is as follows:

Figure 12: An example of a user's initial requirement

XML to Interchange Multidimensional Properties

One key aspect in the success of an MD model should be its capability to interchange information in an easy and standard format. The extensible Markup Language (XML) (W3C, 2000) is rapidly being adopted as the standard for the exchange of un-structured, semi-structured and structured data.

Furthermore, XML is an open neutral platform and vendor independent meta-language, which allows users to reduce the cost, complexity, and effort required in integrating data within and between enterprises. In the future, all applications may exchange their data in XML, and therefore, conversion utilities will not be necessary any more.

We have adopted the XML to represent our MD models due to its advantages, such as standardization, usability, versatility and so on. We have defined a Document Type Definition (DTD) that determines the correct structure and content of XML documents that represent MD models. Moreover, this DTD can be used to automatically validate the XML documents. In Appendix 1 we include the whole DTD that we have defined to represent MD models in XML. This DTD allows us to represent both structural and dynamic properties of MD models.

Table 2: DTP main rules

In Table 2, we have summarized the main rules of our DTD, which contains 38 elements (tags). We have defined additional elements (in plural form) in order to group common elements together, so that they can be exploited to provide optimum and correct comprehension of the model, e.g., elements in plural like PKSCHEMAS or DEPENDENCIES.

The DTD follows the three-level structure of our MD approach:

* An MD model contains PKSCHEMAS (star schema packages) at level 1 (Table 2, line 1).

* A PKSCHEMA contains at most one PKFACT (fact package) and many PKDIMS (dimension packages) and IMPPKDIMS (imported dimensions) at level 2 (Table 2, line 2).

* A PKFACT contains at most one FACTCLASS (Table 2, line 4) and a PKDIM contains at most one DIMCLASS and many BASECLASSES (Table 2, line 3) at level 3.

Within our DTD, fact classes labeled FACTCLASS may have no fact attributes to consider factless fact tables, as can be observed in the content of the element FACTATTS (Table 2, line 6): 0 or more FACTATT.

From now on, we are going to explain the structure of our DTD by means of the grocery example in the previous section. In the next fragment of the XML document that represents the grocery example, the first line defines the XML version and the character encoding used in the document. The next line declares the DTD that defines the structure of the document. Finally, the third line describes the root element of the document (MDMODEL). An MD model (Table 2, line 1) contains star schema packages (PKSCHEMAS) with dependencies between them (DEPENDENCIES) and users' initial requirements (CUBE CLASSES).
In this example, the MD model only contains one star schema package (Figure 7); as there is not any dependency between star schema packages, the DEPENDENCIES element is empty. Finally, the CUBECLASSES element is also empty as no initial requirement has been specified yet.

In our DTD, every MD element has an ID attribute that must be unique to the whole XML document. The value of this attribute is automatically generated by our exportation process and is used in the definition of relationships between elements in our MD model, e.g., in the definition of the dependencies between packages.

The next fragment represents the definition of the star schema Grocery. A PKSCHEMA (Table 2, line 2) can contain:

* At most one PKFACT.

* 0 or more dimension packages (PKDIMS) defined in the very star schema.

* 0 or more dimension packages imported form other star schemas (IMPPKDIMS).

* Dependencies between the dimensions packages (DEPENDENCIES).

Every package, regardless of being a fact or a dimension package, has a name used in the exportation process and a caption used in the graphical representation. As seen in this fragment of the XML document, two dependencies have been defined from the Sales products package (ID7) to the Product package (ID8) and the Time package (ID9). Thanks to the use of the IDREF attribute type in the DTD, we can define that the Start and end attributes of the DEPENDENCY element must take a value from an ID attribute of an element in the XML document.

The following fragment defines the dimension package Product. A dimension package (Table 2, line 3) contains:

* At most one dimension class (DIMCLASS).

* 0 or more base classes that represent hierarchy levels (BASECLASSES).

* 0 or more imported bases classes from other dimension packages (IMPBASECLASSES).

In this fragment we can see how the relationships between a dimension class and base classes are expressed in our DTD; the cardinality of the relationship is expressed by means of the attributes roleA and roleB. We can also see the definition of the three attributes of the Product dimension class: upc, name, and weight. In the DTD, the {OID} and {D} constraints of our MD model are represented as attributes OID and D of the DIMATT element.

CASE STUDIES

The aim of this section is to exemplify the usage of our conceptual modeling approach on modeling MD databases. We have selected three different examples taken from Kimball’s book (Kimball, 2002), each of which introduces a new particular modeling feature: a warehouse, a large bank, and a college course. Due to the lack of space, we will only apply our complete modeling approach for the first example: we will apply all of the diagrams we use for modeling a DW (package diagrams, class diagrams, interaction diagrams, etc.). For the rest of the examples, due to space constraints, we will only focus on representing the structural properties of MD modeling by specifying the corresponding UML class diagram. This class diagram is the key one in our approach since the rest of the diagrams can be easily obtained from it.

The Warehouse

This example explores three inventory models of a warehouse. The first one is the inventory snapshot, where the inventory levels are measured every day and are placed in separate records in the database. The second model is the delivery status model, which contains one record for each delivery to the warehouse and the disposition of all the items is registered until they have left the warehouse. Finally, the third inventory model is the transaction model, which records every change of the status of delivery products as they arrive at the warehouse, are processed into the warehouse,

etc.

This example introduces two important concepts: the semi-additivity and the multistar model (also known as fact constellations). The former has already been introduced and refers to the fact that a measure cannot be summarized by using the sum function along a dimension. In this example, the inventory level (stock) of the warehouse is semi-additive, because it cannot be summed along the time dimension, but it can be averaged along the same dimension. The multistar (fact constellations) concept refers to the fact that the same MD model has multiple facts.

To start with, in our approach we model multistar models by means of package diagrams. In this way, at the first level, we create a package diagram for each one of the facts considered in the model. At this level, connecting package diagrams means that a model will use elements (e.g., dimensions, hierarchies) defined in the other package. Figure 13 shows the first level of the model formed by three packages that represent the different star schemas in the case study.

Then, we explore each package diagram at the second level to define packages for each one of the facts and dimensions defined in the corresponding package diagram. Figure 14 shows the content of the package Inventory Snapshot Star at level 2. The fact package Inventory Snapshot Fact is represented in the middle of Figure 14, and the dimension packages (Product Dimension, Time Dimension, and Warehouse Dimension) are placed around the fact package. As noted, a dependency is defined from the fact package to each one of the dimension packages, because the fact package comprises the whole definition of the star schema. At level 2, it is possible to create a dependency from a fact package to a dimension package or between dimension packages (when they share some hierarchy levels), but not from a dimension package to a fact package.

Figure 13: Level 1 of the Warehouse case study

Figure 15 shows the content of the package Inventory Transaction Star at level 2. As in the Inventory Snapshot Star, the fact package is placed in the middle of the figure and the dimension packages are placed around the fact package in a star fashion. Three dimension packages (Product Dimension, Time Dimension, and Warehouse Dimension) have been previously defined in the Inventory Snapshot Star (Figure 14), and they are imported in this package. Therefore, the name of the package where it has been previously defined appears below the package name (from Inventory Snapshot Star).

The content of the dimension and fact packages is represented at level 3. The diagrams at this level are only comprised of classes and their associations. For example, Figure 16 shows the content of the package Warehouse Dimension at level 3. In a dimension package, a class is specified for the dimension class (Warehouse) and a class for each classification hierarchy level (ZIP, City, County, State, SubRegion, and Region). For the sake of simplicity, the methods of each class have not been depicted in the figure. As can be seen in Figure 16, Warehouse presents alternative path classification hierarchies: (i) ZIP, City, County, State, and (ii) SubRegion, Region, State.

Figure 14: Level 2 of the Inventory Snapshot Star

Figure 75: Level 2 of the Inventory Transaction Star

Figure 16: Level 3 of Warehouse Dimension

Finally, Figure 17 shows the content of the package Inventory Snapshot

Fact, In this package, the whole star schema is displayed: the fact class (Inventory Snapshot) is defined and the dimensions with their corresponding hierarchy levels are imported from the dimension packages. To avoid unnecessary details, we have hidden the attributes and methods of dimensions and hierarchy levels, but the measures of the fact are shown as attributes of the fact class: four atomic measures (quantity_on_hand, quantity_shipped, value_at_cost, and value_at_LSP), and three derived measures (number_of_turns, gross_profit, and gross_margin). The definition of the derived measures is included in the model by means of derivation rules. Regarding the additivity of the measures, only quantity_on_hand is semiadditive; and therefore, an additivity rule has been added to the model. Finally, Warehouse presents alternative path classification hierarchies and Time and Product present multiple classification hierarchies, as can be seen in Figure 17.

Figure 17: Level 3 of the Inventory Snapshot Fact

Regarding the dynamic part of the model, let us suppose the following user's initial requirement on the MD model specified by the UML class diagram of Figure 17: "We wish to analyze the quantity _on_hand of products where the group of products is 'Grocery' and the warehouse state is 'Valencia', grouped according to the product subgroup and the warehouse region and subregion, and ordered by the warehouse subregion and region." On the left hand side of Figure 18, we can observe the graphical notation of the cube class that corresponds to this requirement. The measure to be analyzed (quantity_on_hand) is specified in the measure area. Constraints defined on dimension classification hierarchy levels (group and state) are included in the slice area, while classification hierarchy levels along which we are interested in analyzing measures (subgroup, region, and subregion) are included in the dice area. Finally, the available OLAP operations are specified in the CO (Cube Operations) section (in this example the CO are omitted to avoid unnecessary detail). On the right hand side of Figure 18, the OQL sentence corresponding to the cube class is shown. We can notice how the descriptor attributes from the MD model are used when the attributes of the hierarchy levels are omitted in the analysis. For example, the expression Warehouse.State= "Valencia" of the cube class involves the use of the descriptor attribute from the State base class (Figure 16).

Regarding state diagrams, one state diagram is defined for each initial cube class. The diagram specifies that certain OLAP operations lead users to cube classes that allow them to analyze the same data (the same measures along the same dimensions) in different ways. For example, in Figure 19 we can see the corresponding state diagram of the cube class definition of Figure 18. It may be observed, for example, that roll-up and drill-down operations applied on the classification hierarchies levels defined on the Warehouse and Product dimensions will allow us to navigate up and down along the classification hierarchies defined in both dimensions.

On the other hand, an interaction diagram can also be defined for each UML class diagram. This interaction diagram shows interactions among cube classes, changed by OLAP operations such as rotate, pivot, slice, or dice. In Figure 20, we can see an example of an interaction diagram, in which we have considered three cube classes that specify the user's initial requirements. We have then defined the OLAP operations needed to switch between these cube classes.

Figure 18: An example of a user's initial requirement for the Warehouse case study

A Large Bank

In this example, a DW for a large bank is presented. The bank offers a significant portfolio of financial services: checking accounts, savings accounts, mortgage loans, safe deposit boxes, and so on.

This example introduces the following concepts:

* Heterogeneous dimension: a dimension that describes a large number of heterogeneous items with different attributes. Kimball’s recommended technique is "to create a core fact table and a core dimension table in order to allow queries to cross the disparate types and to create a custom fact table and a custom dimension table for querying each individual type in depth". However, thanks to the categorization of dimensions, our conceptual MD approach can provide an elegant and simple solution to this problem.

Figure 19: An example of the state diagram for the Warehouse case study
* Categorization of dimensions: This allows us to model additional features for a dimension’s subtypes.

Figure 20: An example of interaction diagram

* Shared classification hierarchies between dimensions: our approach allows two or more dimensions to share some levels of their classification hierarchies.

Figure 21 represents level 1, which comprises five star packages: Saving Accounts Star, Personal Loans Star, Investment Loans Star, Safe Deposit Boxes Star, and Mortgage Loans Star. From now on, we will only center on the Mortgage Loans Star. The corresponding level 2 of this star package is depicted in Figure 22.

Figure 21: Level 1 of the Bank case study

Figure 22: Level 2 of Mortgage Loans Star

Level 3 of Mortgage Loans Fact is shown in Figure 23. To avoid unnecessarily complicating the figure, three of the dimensions (Account, Time, and Status) with their corresponding hierarchies are not represented. Moreover, the attributes of the represented hierarchy levels have been omitted. The fact class (Mortgage Loans) contains four attributes that represent the measures: total, balance, and payment_number are atomic; whereas debt is derived (the corresponding derivation rule is placed next to the fact class). None of the measures is additive. Consequently, the additivity rules are also placed next to the fact class.

In this example, the dimensions present two special characteristics. On one hand, Branch and Customer share some hierarchy levels: ZIP, City, County, and State. On the other hand, the Product dimension has a generalization-specialization hierarchy. This kind of hierarchy allows us to easily deal with heterogeneous dimensions: the different items can be grouped together in different categorization levels depending on their properties.

The College Course

This example introduces the concept of the factless fact table (FFT): fact tables for which there are no measured facts. Kimball distinguishes two major variations of an FFT: event tracking tables and coverage tables. In this example we will focus on the first type.

Event tracking tables are used when a large number of events need to be recorded as a number of dimensional entities coming together

simultaneously. In this example, we will model daily class attendance at a college. In Figure 24 and Figure 25, level 1 and level 2 of this model are depicted respectively. In this case, level 1 only contains one star package.

Figure 26 shows level 3 of College Course Fact. For the sake of simplicity, the attributes and methods of every class have not been depicted in the figure. As shown, the fact class College Course contains no measures because it is an FFT. In FFT, the majority of the questions that users create imply counting the number of records that satisfy a constraint, such as which facilities were used most heavily. Or, which courses were the least attended?

Figure 23: Level 3 of the Mortgage Loans Fact

Figure 24: Level lof the College case study

Regarding the dimensions, Course and Time present multiple classification hierarchies, Professor and Student share some hierarchy levels, and Facility presents a categorization hierarchy.

CONCLUSIONS

In this paper, we have presented an OO conceptual modeling approach, based on the UML, to design DWs, MD databases and OLAP applications. Structural aspects of MD modeling are easily specified by means of a UML class diagram in which classes are related through association and shared aggregation relationships. In this context, thanks to the flexibility and the power of the UML, all the semantics required for proper MD conceptual modeling are considered, such as many-to-many relationships between facts and particular dimensions, multiple path hierarchies of dimensions, the strictness and completeness of classification hierarchies, and categorization of dimension attributes. Regarding dynamic aspects, we provide a UML-compliant class graphical notation (called cube classes) to specify users' initial requirements at the conceptual level. We have also described how we use state and interaction diagrams to model the behavioral aspects of the system regarding these cube classes based on the set of the applied OLAP operations. Moreover, we have sketched out how to represent a conceptual MD model accomplished by our approach in the ODMG standard as a previous step for a further implementation of MD models into OODB and ORDB. Furthermore, to facilitate the interchange of MD models, we provide a DTD from which we can obtain valid XML documents. Finally, we have selected three case studies from Kimball's book and modeled them following our approach. This shows that our approach is a very easy-to-use yet powerful conceptual model that represents main structural and dynamic properties of MD modeling in an easy and elegant way.

Figure 25: Level 2 of the College Course Star

Figure 26: Level 3 of the College Course Star

Currently, we are working on several issues. On one hand, we are extending our approach to key issues in MD modeling, including temporal and slowly changing dimensions. On the other hand, we are also working on a particular metadata to represent MD properties in object-oriented databases and avoid the loss of expressiveness we have when we transform a conceptual schema accomplished by our approach into the ODMG standard.

See Appendix beginning on next page.

APPENDIX 1

APPENDIX 1

APPENDIX 1

ENDNOTES

1 A descriptor attribute will be used as the default label in the data analysis.

2 See Chapter 4 (Trujillo, 2001a; Trujillo et al., 2000) for more information about the provided OLAP operations and how to build state and interaction diagrams.
3 The ODMG defines an ODBMS as "[...] a DBMS that integrates database capabilities with object-oriented programming language capabilities".

REFERENCES

Abello, A., Samos, J. & Saltor, F. (2001). A framework for the Classification and Description of Multidimensional Data Models. Proceedings of the Twelfth Database and Expert Systems Applications (DEXA'01), Munich, Germany, Lecture Notes in Computer Science (LNCS), 2113, 668-677.

Blaschka, M., Sapia, C., Hofling, G. & Dinter, B. (1998). Finding Your Way Through Multidimensional Models. Proceedings of the Ninth International Workshop on Database and Expert Systems Applications (DEXA'98), Vienna, Austria, IEEE Computer Society, 198-203.

Booch, G., Rumbaugh, J. & Jacobson, I. (1998). The Unified Modeling Language. New York: Addison-Wesley.

Cattell, R. G. G. et al. (2000). The Object Data Standard: ODMG 3.0. New York: Morgan Kaufmann Publishers, 2000.

Chaudhuri, S. & Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology. A CM Sigmod Record, 26(1), 65-74.
Giovinazzo, W. (2000). Object-Oriented Data Warehouse Design. Building a star schema. New York: Prentice-Hall.

Golfarelli, M., Maio, D. & Rizzi, S. (1998). The Dimensional Fact Model: a Conceptual Model for Data Warehouse. International Journal of Cooperative Information Systems, 7(2&3), 215-247.

Golfarelli, M., Rizzi, S. & Vrdoljak, B. (2001). Data warehouse design from XML sources. Proceedings of the ACM Fourth International Workshop on Data Warehousing and OLAP (DOLAPOl), Atlanta, US A, 40-47.

Kimball, R. & Ross, M. (2002). The Data Warehouse Toolkit (2nd ed.). New York: John-Wiley & Sons.

Lehner, W. (1998). Modeling Large Scale OLAP Scenarios. Proceedings of the Sixth International Conference On Extending Database Technology (EDBT'98), Valencia, Spain, Lecture Notes in computer Science (LNCS), 1377, 153-167.

Lujan-Mora, S., Trujillo, J. & Song, L-Y. (2002). Multidimensional Modeling with UML Package Diagrams". Proceedings of the Twenty-first international Conference on Conceptual Modeling (ER'02), Tampere, Finland, Lecture Notes in Computer Science, 2503,199-213.

Object Management Group (OMG) (2001). Unified Modeling Language Specification 1.4. [Online]. Available: http://www.omg.org.

Pokorny, J. (2001). Modelling Stars Using XML. Proceedings of the ACM Fourth International Workshop on Data Warehousing and OLAP (DOLAPOl),

Atlanta, USA, 24-31. {dot}

Sapia, C., Blaschka, M., Hofling, G. & Dinter, B. (1998). Extending the E/R Model for the Multidimensional Paradigm. ER Workshops 7998, Singapore, Lecture Notes in Computer Science (LNCS), 1552, 105-116.

Sapia, C. (1999). On Modeling and Predicting Query Behavior in OLAP Systems. Proceedings of the First International Workshop on Design and Management of Data Warehouse (DWDM'99), Heidelberg, Germany.

Trujillo, J., Palomar, M. & Gomez, J. (2000). Modeling the Behavior of OLAP Applications Using an UML Compliant Approach. Proceeedings of the First Conference on Advances in Information Systerna (ADVlSOO). Izmir, Turin Computer Science (LNCS), 1909, 14-23.

Trujillo, J (200Ia). The GOLD model: an object oriented conceptual model for the design of OLAP applications. Unpublished doctoral dissertation, Universidad de Alicante, Alicante, Spain.

Trujillo, J., Palomar, M., Gomez, J. & Song, I.-Y. (200Ib). Designing Data Warehouses with OO conceptual models. IEEE Computer, special issue on Data Warehouses, 34 (12), 66-75.

Tryfona, N., Busborg, F. & Christiansen, J.G.B. (1999). starER: A Conceptual Model for Data Warehouse Design. Proceedings of the ACM second International Workshop on Data Warehousing and OLAP (DOLAP'99), Kansas City, USA, 3-8.

World Wide Web Consortium (W3C) (2000, October). Extensible Markup Language (XML) 1.0 (2nd ed.). [Online]. Available: http://www.w3.org/TR/2000/REC-xml-20001006.

Warmer, J. & Kleppe, A. (1998). The Object Constraint Language: Precise Modeling with UML. New York: Addison-Wesley.

Juan Trujillo and Sergio Lujan-Mora, Universidad de Alicante, Spain Il-Yeol Song, Drexel University, USA

Juan Trujillo is a professor at the Computer Science School at the University of Alicante, Spain. He received a Ph.D. in Computer Science from the University of Alicante (Spain) in 2001. His research interests include database modeling, conceptual design of data warehouses, multidimensional databases, OLAP, and object oriented analysis and design with UML. He has published papers in international conferences and journals such as ER, UML, ADBIS, WAiM and IEEE Computer. He is a Program Committee member of several workshops and conferences such as ER, DOLAP, and SCI. Contact him at jtrujillo@dlsi.ua.es.

Sergio Lujan-Mora is a Lecturer at the Computer Science School at the University of Alicante, Spain. He received a BS in 1997 and a Master in Computer Science in 1998 from the University of Alicante. Currently he is a doctoral student in the Dept. of Language and Information Systems being advised by Dr. juan Trujillo. His research spans the fields of Multidimensional Databases, Data Warehouses, OLAP techniques, Database Conceptual Modeling and Object Oriented Design and Methodologies, Web engineering and Web programming. he is author of several national journal papers and papers presented in international conferences such as ER, UML and WAlM. Contact him at sergio.lujan@ua.es.

Il-Yeol Song is a professor at Drexel University. he received a Ph.D. in Computer Science from Louisiana State University, Baton Rouge, LA, in 1988. His research interests include database modeling, design & performance

optimization of data warehouses, OLAP, database support for e-commerce systems, and object-oriented analysis & design with UML. he has published over 80 refereed papers in conferences and journals. He served as a program committee member of over 60 conferences and workshops. he served as a program co-chair for DOLAP 98, CIKM99, DOLAP99, and ER2003. he received three teaching awards from Drexel University including Exemplary Teaching Award (1992), Teaching Excellence Award (2000), and the most prestigious Lindback Distinguished Teaching Award (2001). Contact him at songiy@drexel.edu.

THIS IS THE FULL-TEXT.

Descriptors: Studies; Data base management; Data warehouses; Extensible Markup Language; Object oriented programming; Online analytical processing
Classification Codes: 9130 (CN=Experimental/Theoretical); 5240 (CN=Software & systems)
Print Media ID: 11186

Abstract:

...by the UML to simplify the final model. Furthermore, a UML-compliant class notation (called cube class) is provided to represent OLAP users' initial requirements. The paper also describes how to...

Text:

...UML to simplify the final model. Furthermore, we provide a UML-compliant class notation (called cube class) to represent OLAP users' initial requirements. We also describe how we can use the...

...we have provided a UML-compliant class notation to represent OLAP users' initial requirements (called cube class). From these cube classes, we then describe the use of state and interaction diagrams to model the behavior...The next section details the major features of MD modeling that should be taken into account for a proper MD conceptual design. Then the paper summarizes the most relevant conceptual approaches...

...fact attributes (atomic or derived), which are contained in cells or points in the data cube . This set of measures is based on a set of dimensions that determine the granularity...

...and time. On the left hand side of Figure 1, we can observe a data cube typically used for representing an MD model. In this particular case, we have defined a cube for analyzing measures along the product, store and time dimensions.

We note that a fact...
...OLAP terminology), however, might not be semantically meaningful for all measures along all dimensions. A measure is semi - additive if the SUM operator can be applied to some dimensions, but not all the dimensions...

...additive measures would be those measures that record a static level

such as inventory, financial account balances or measures of intensity such as room temperatures (Kimball and Ross, 2002).

Figure 1: A data cube and classification hierarchies defined on dimensions

Regarding dimensions, the classification hierarchies defined on certain dimension...number of dimensions increases significantly, which may then lead to extremely sparse dimensions and data cubes . In this way, there are some attributes that are normally valid for all elements within... diagram. Secondly, we summarize how users' initial requirements are easily considered by what we call cube classes. We next describe how to model the behavior of MD databases by using UML state and interaction diagrams from the information represented in these cube classes. The next section sketches how we automatically transform an MD model accomplished by following...

...us to specify users' initial requirements by means of a UML-compliant class notation called cube class. After requirements are specified, behavioral properties are usually then related to these cube classes that represent users' initial requirements. We particularly use state and interaction diagrams to model the behavior (evolution) of these cube classes based on the applied OLAP operation.

Cube classes follow the query-by-example (QBE) method: the requirements are defined by means of...QBE systems are considered easier to learn than formal query languages. The structure of a cube class is shown in Figure 4:

* Cube class name.

* Measures area, which contains the measures from the fact to be analyzed.

* Slice...

...conditions to address the analysis.

* Order area, which specifies the order of the result set.

* Cube operations, which cover the OLAP operations for a further data-analysis phase.

We should point out that this graphical notation of the cube class aims at facilitating the definition of users' initial requirements to non-expert UML or databases users. In a more formal way, every one of these cube classes has its underlying OQL specification. Moreover, an expert user can directly define cube classes by specifying the OQL sentences (see more details on the representation of cube classes further in the paper).

Figure 4: Cube class structure

Behavioral Properties by Using State and Interaction Diagrams

From these cube classes seen in the previous section, final users may start a navigational process by applying...

...etc.) in the further data analysis phase. These operations are closed as they generate another cube class as an output. Thus, we use state and interaction diagrams2 to model the behavior (evolution) of these cube classes based on the applied OLAP operation. These diagrams contain information about the most probable...

...proper view maintenance policy.

Regarding state diagrams, one state diagram is defined for each initial cube class. The diagram specifies that certain OLAP operations lead users to cube classes that allow them to analyze the same data (the same measures along the same...

...considered as a valid state. Every one of these valid states will be a new cube class. Then, the provided OLAP operations allow us to navigate along the states to define new cube classes.

In Figure 5, we can see an example of state diagrams. One state is...

...in the classification hierarchy of the dimensions included in the corresponding Dice area of the cube class. The data analysis will start on the initial state that corresponds to the finest...

...1998; OMG, 2001) for their clarity and low complexity. This interaction diagram shows interactions among cube classes, changed by OLAP operations such as rotate, pivot, slice, or dice. Thus, we can specify that certain OLAP operations (e.g., dice) lead users to cube classes which will show completely different data. Thus, these new cube classes represent the most probable new requirements a final user wishes to execute.

In Figure...

...see an example of interaction diagrams. Let us suppose that we have only defined two cube classes to specify two initial requirements. Then, we specify the operation needed to switch from one cube class into the other. In this particular case, the rotate operation indicates the transition that...

...an MD model in most of the relational database management systems such as Oracle, Informix, Microsoft SQL Server, IBM DB2 and so on (Trujillo et al., 2001b). Furthermore, we also provide...that cannot be directly represented by using this standard and that should be taken into account when transforming this ODBM into a particular object-oriented model of the target ODBMS.

The...

...used to define the specifications of object types. Then, we will sketch how to represent cube classes into the OQL, a query language that supports the ODMG data model. The great...should be used. Unfortunately, a constraint language is completely missing from the ODMG standard specification.

Cube Classes Represented by Using OQL

The OQL is not easy to use for defining users...

...underlying ODL representation corresponding to the MD model. Due to this fact, we also provide cube classes, which allow the user to define initial requirements in a graphical way. These cube classes can automatically be transformed into OQL sentences, and can therefore be used to query...

...ordered by the Brand of the product

In Figure 12, we can see the corresponding cube class to the previous requirement. It is easy to see how the cube class is formed:

* Measures contains the goal of the analysis: SUM(quantity).

* Slice the restrictions...

...the Product and Time dimensions.

* And Order defines the order of the result set.

The cube class can be automatically translated into OQL. The algorithm uses the corresponding ODL definition of...

...year in Month base class. Moreover, when attributes' names are omitted in the cube class, the algorithm automatically selects the descriptor attribute defined in the MD model. For example, the expression Time.Month= "January" of the cube class in Figure 12 involves the use of the descriptor attribute from the Month base...

...Brand involves the use of the descriptor attribute from Brand. The OQL for the corresponding cube class in Figure 12 is as follows:

Figure 12: An example of a user's...1) contains star schema packages (PKSCHEMAS) with dependencies between them (DEPENDENCIES) and users' initial requirements (CUBE CLASSES).

In this example, the MD model only contains one star schema package (Figure 7...

...is not any dependency between star schema packages, the DEPENDENCIES element is empty. Finally, the CUBECLASSES element is also empty as no initial requirement has been specified yet.

In our DTD...the left hand side of Figure 18, we can observe the graphical notation of the cube class that corresponds to this requirement. The measure to be analyzed (quantity...

...included in the dice area. Finally, the available OLAP operations are specified in the CO (Cube Operations) section (in this example the CO are omitted to avoid unnecessary detail). On the right hand side of Figure 18, the OQL sentence corresponding to the cube class is shown. We can notice how the descriptor attributes from the MD model are...

...levels are omitted in the analysis. For example, the expression Warehouse.State= "Valencia" of the cube class involves the use of the descriptor attribute from the State base class (Figure 16).

Regarding state diagrams, one state diagram is defined for each initial cube class. The diagram specifies that certain OLAP operations lead users to cube classes that allow them to analyze the same data (the same measures along the same...

...ways. For example, in Figure 19 we can see the corresponding state diagram of the cube class definition of Figure 18. It may be observed, for example, that roll-up and...

...can also be defined for each UML class diagram. This interaction diagram shows interactions among cube classes, changed by OLAP operations such as rotate, pivot, slice, or dice. In Figure 20, we can see an example of an interaction diagram, in which we have considered three cube classes that specify the user's initial requirements. We have then defined the OLAP operations needed to switch between these cube classes.

Figure 18: An example of a user's initial requirement for the Warehouse

case...
...a large bank is presented. The bank offers a significant portfolio of financial services: checking accounts , savings accounts , mortgage loans, safe deposit boxes, and so on.

This example introduces the following concepts:

* Heterogeneous...

...of their classification hierarchies.

Figure 21 represents level 1, which comprises five star packages: Saving Accounts Star, Personal Loans Star, Investment Loans Star, Safe Deposit Boxes Star, and Mortgage Loans Star...

...is shown in Figure 23. To avoid unnecessarily complicating the figure, three of the dimensions (Account , Time, and Status) with their corresponding hierarchies are not represented. Moreover, the attributes of the...of dimension attributes. Regarding dynamic aspects, we provide a UML-compliant class graphical notation (called cube classes) to specify users' initial requirements at the conceptual level. We have also described how...

...use state and interaction diagrams to model the behavioral aspects of the system regarding these cube classes based on the set of the applied OLAP operations. Moreover, we have sketched out...

? b 411

```
       24feb10 08:24:25 User276702 Session D274.3
            $3.51    0.599 DialUnits File15
               $1.91  1 Type(s) in Format  3
               $4.09  1 Type(s) in Format  9
            $6.00  2 Types
     $9.51  Estimated cost File15
            $3.55    0.655 DialUnits File324
               $1.95  1 Type(s) in Format  3
            $1.95  1 Types
     $5.50  Estimated cost File324
            $4.80    0.931 DialUnits File349
               $1.70  1 Type(s) in Format  3
            $1.70  1 Types
     $6.50  Estimated cost File349
           $26.74    4.185 DialUnits File654
               $1.64  2 Type(s) in Format  3
            $1.64  2 Types
    $28.38  Estimated cost File654
            $1.97    0.425 DialUnits File996
     $1.97  Estimated cost File996
            OneSearch, 5 files,  6.796 DialUnits FileOS
     $1.34  INTERNET
    $53.20  Estimated cost this search
   $190.76  Estimated total session cost  50.310 DialUnits
File 411:DIALINDEX(R)
```

```
DIALINDEX(R)
   (c) 2010 Dialog

*** DIALINDEX search results display in an abbreviated ***
*** format unless you enter the SET DETAIL ON command. ***
```

? **sf all**

```
   You have 508 files in your file list.
   (To see banners, use SHOW FILES command)
```

? **s ((semiadditive? or (semi () additive)) and (non () additive)) (n2) (measure? or metric? or value? or parameter?) and (cube? and account?) not py>2004**

Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing

Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing

```
Your SELECT statement is:
   S  ((SEMIADDITIVE? OR (SEMI () ADDITIVE)) AND (NON () ADDITIVE)) (N2)
(MEASURE? OR METRIC? OR VALUE? OR PARAMETER?) AND (CUBE? AND ACCOUNT?) NOT
PY>2004

           Items   File
           -----   ----
               1    15: ABI/Inform(R)_1971-2010/Feb 23
       Examined  50 files
       Examined 100 files
       Examined 150 files
       Examined 200 files
               1   349: PCT FULLTEXT_1979-2010/UB=20100205|UT=20100204
       Examined 250 files
       Examined 300 files
       Examined 350 files
       Examined 400 files
Processing
               3   654: US PAT.FULL._1976-2010/FEB 11
       Examined 450 files
       Examined 500 files
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
               1   996: Newsroom 2004_

   4 files have one or more items; file list includes 508 files.
   One or more terms were invalid in 66 files.
```

? b hits

```
       24feb10 08:35:19 User276702 Session D274.4
           $92.43   29.912 DialUnits File411
     $92.43  Estimated cost File411
      $2.94  INTERNET
     $95.37  Estimated cost this search
    $286.13  Estimated total session cost  80.222 DialUnits

SYSTEM:OS  - DIALOG OneSearch
  File  15:ABI/Inform(R) 1971-2010/Feb 23
         (c) 2010 ProQuest Info&Learning
  File 349:PCT FULLTEXT 1979-2010/UB=20100205|UT=20100204
         (c) 2010 WIPO/Thomson
  File 654:US PAT.FULL. 1976-2010/FEB 11
         (c) Format only 2010 Dialog
*File 654: Reassignment data for 2009 delayed due to technical issues.
See File 123 for current reassignments.
  File 996:Newsroom 2004
         (c) 2009 Dialog

      Set  Items  Description
      ---  -----  -----------
```

? s ((semiadditive? or (semi () additive)) and (non () additive)) (n2) (measure? or metric? or value? or parameter?) and (cube? and account?) not py>2004

Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Processing

```
Processing
Processing
Processing
Processing
Processing
            215  SEMIADDITIVE?
        1014001  SEMI
         415381  ADDITIVE
           1993  SEMI(W)ADDITIVE
        5682267  NON
         415381  ADDITIVE
           1528  NON(W)ADDITIVE
        4449868  MEASURE?
         304933  METRIC?
        5844699  VALUE?
        1804592  PARAMETER?
             10  ((SEMIADDITIVE? OR SEMI(W)ADDITIVE) AND
                 NON(W)ADDITIVE)(2N)(((MEASURE? OR METRIC?) OR VALUE?) OR
```

```
                PARAMETER?)
        127419  CUBE?
       3199092  ACCOUNT?
       5619695  PY>2004
    S1       6  ((SEMIADDITIVE? OR (SEMI () ADDITIVE)) AND (NON ()
                ADDITIVE)) (N2) (MEASURE? OR METRIC? OR VALUE? OR
                PARAMETER?) AND (CUBE? AND ACCOUNT?) NOT PY>2004
```

? **rd**

```
>>>Duplicate detection is not supported for File 349.

>>>Duplicate detection is not supported for File 654.

>>>Records from unsupported files will be retained in the RD set.
    S2       5  RD  (unique items)
```

? **t s2/ti/all**

Dialog eLink: USPTO Full Text Retrieval Options

2/TI/1 (Item 1 from file: 15)
DIALOG(R)File 15: ABI/Inform(R)

Applying UML and XML for designing and interchanging information for data warehouses and OLAP applications

2/106/2 (Item 1 from file: 349)
DIALOG(R)File 349: PCT FULLTEXT

846386

ANALYTICAL SERVER INCLUDING METRICS ENGINE
SERVEUR ANALYTIQUE COMPRENANT UN MOTEUR DE MESURES

2/TI/3 (Item 1 from file: 654)
DIALOG(R)File 654: US PAT.FULL.

Analytical server including metrics engine

2/TI/4 (Item 2 from file: 654)
DIALOG(R)File 654: US PAT.FULL.

Analytical server including metrics engine

2/TI/5 (Item 3 from file: 654)
DIALOG(R)File 654: US PAT.FULL.

Analytical server including metrics engine

? **t s2/3,k/5**

Dialog eLink: Order File History
2/3,K/5 (Item 3 from file: 654)
DIALOG(R)File 654: US PAT.FULL.


```
0005004605 **IMAGE Available
Derwent Accession: 2002-205726
```
Analytical server including metrics engine
```
Inventor: Arun Shah, INV
          Robert Novy, INV
Correspondence Address: Brian C. McCormack Baker & McKenzie, 2001 Ross
    Avenue, Suite 2300, Dallas, TX, 75201, US
```

	Publication Number	Kind	Date	Application Number	Filing Date
Main Patent	US 20020059267	A1	20020516	US 2001837114	
20010417					
Provisional				US 60-197894	20000417
Provisional				US 60-199975	20000427

Fulltext Word Count: 9525

**IMAGE Available

Description of the Invention:
...metrics, and returns them to the clients in a structured form such as a multidimensional cubes .

[...

...in the time dimension. To accommodate queries of measure across different dimensions and to properly account for the varying additive/non - additive properties of measures in different dimensions, a second "additivity" flag may be provided at 416, which might for...

...The use of these additivity flags will be discussed in the section below relating to non - additive measures and metrics .

[...0060] C. Non - additive Metric Calculation...

...simply total the returned results. However, this is incorrect wherein certain measure components of the metrics are non - additive . Correct totals can only be obtained if the requester has knowledge of which measures are non - additive and asks for the non - additive measures separately...

...to FIG. 4, it is possible for the analytical server 120 to readily determine which measures are non - additive . By making this determination, the analytical server can allow the rollup to be handled transparently...

...hierarchical levels of the aggregate tables 130 allows for an efficient implementation of the transparent non - additive metric calculations described above...

...0064] At step 820, a separate totals query is generated for each non - additive measure . The query is launched using the stars as described above, and it is noted that...

...is terminated. In the foregoing manner, complex metrics composed of any combination of additive and non - additive measures can be calculated correctly and efficiently, without requiring any knowledge or action on the part...

...example, using the additive/non-additive fields 411,416, the analytical server 120 knows which measures are non - additive . Accordingly, the handling of the non - additive measures can be handled transparently, without making the non-additive attribute visible to the requester. This ...

...monthly sales of product by business unit, the request would be for a three-dimensional cube (business unitXmonthXmetric values). If the sales were not additive across the product dimension a separate...

...with the values representing the totals across all business unit. Alternatively, the original three-dimensional cube might simply reserve one extra element in the first dimension to contain the totals...

...0068] When a measure is non - additive , the analytical server 120 instead generates and issues two separate queries, the extra query being ...

...for Business Unit). In this way, complex metrics composed of any combination of additive and non - additive measures can be calculated correctly and efficiently, without requiring any knowledge or action on the part...

Exemplary or Independent Claim(s):
...additive, and wherein the metadata includes a designation of which measures are additive and which measures are non - additive , the method comprising: (a) receiving in the analytical server, from the RDBMS, at least a...

Non-exemplary or Dependent Claim(s):
...of claim 12 wherein the determining of the at least one database query takes into account whether the requested metric is additive

```
specifically across the requested dimension...
```

?